

Doxxygen_MPDASTARONLINE_DB Reference Manual

Generated by Doxygen 1.3.7

Wed Sep 6 11:53:35 2006

Contents

1	Doxygen_MPDAUTHENTICATION Class Index	1
1.1	Doxygen_MPDAUTHENTICATION Class List	1
2	Doxygen_MPDAUTHENTICATION File Index	3
2.1	Doxygen_MPDAUTHENTICATION File List	3
3	Doxygen_MPDAUTHENTICATION Class Documentation	5
3.1	ftpcAnodesSender Class Reference	5
3.2	ftpcCathodeSender Class Reference	16
3.3	ftpcGasSystemSender Class Reference	26
3.4	ftpcGGridSender Class Reference	37
3.5	ftpcHDLCTempsSender Class Reference	47
3.6	ftpcHDLCVoltagesSender Class Reference	58
3.7	ftpcTempsSender Class Reference	69
4	Doxygen_MPDAUTHENTICATION File Documentation	81
4.1	ftpcAnodesDaemon.cc File Reference	81
4.2	ftpcAnodesSender.cc File Reference	82
4.3	ftpcAnodesSender.hh File Reference	83
4.4	ftpcAnodesSender_i.cc File Reference	84
4.5	ftpcCathodeDaemon.cc File Reference	85
4.6	ftpcCathodeSender.cc File Reference	86
4.7	ftpcCathodeSender.hh File Reference	87
4.8	ftpcCathodeSender_i.cc File Reference	88
4.9	ftpcGasSystemDaemon.cc File Reference	89
4.10	ftpcGasSystemSender.cc File Reference	90
4.11	ftpcGasSystemSender.hh File Reference	91
4.12	ftpcGasSystemSender_i.cc File Reference	92
4.13	ftpcGGridDaemon.cc File Reference	93

4.14	ftpcGGridSender.cc File Reference	94
4.15	ftpcGGridSender.hh File Reference	95
4.16	ftpcGGridSender_i.cc File Reference	96
4.17	ftpcHDLCTempsDaemon.cc File Reference	97
4.18	ftpcHDLCTempsSender.cc File Reference	98
4.19	ftpcHDLCTempsSender.hh File Reference	99
4.20	ftpcHDLCTempsSender_i.cc File Reference	100
4.21	ftpcHDLCVoltagesDaemon.cc File Reference	101
4.22	ftpcHDLCVoltagesSender.cc File Reference	102
4.23	ftpcHDLCVoltagesSender.hh File Reference	103
4.24	ftpcHDLCVoltagesSender_i.cc File Reference	104
4.25	ftpcTempsDaemon.cc File Reference	105
4.26	ftpcTempsSender.cc File Reference	106
4.27	ftpcTempsSender.hh File Reference	107
4.28	ftpcTempsSender_i.cc File Reference	108

Chapter 1

Doxxygen_MP_D_STAR_ONLINE_DB Class Index

1.1 Doxygen_MP_D_STAR_ONLINE_DB Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ftpcAnodesSender	5
ftpcCathodeSender	16
ftpcGasSystemSender	26
ftpcGGridSender	37
ftpcHDLCTempsSender	47
ftpcHDLCVoltagesSender	58
ftpcTempsSender	69

Chapter 2

Doxxygen_MPDASTARONLINE_DB File Index

2.1 Doxygen_MPDASTARONLINE_DB File List

Here is a list of all files with brief descriptions:

ftpcAnodesDaemon.cc	81
ftpcAnodesSender.cc	82
ftpcAnodesSender.hh	83
ftpcAnodesSender_i.cc	84
ftpcCathodeDaemon.cc	85
ftpcCathodeSender.cc	86
ftpcCathodeSender.hh	87
ftpcCathodeSender_i.cc	88
ftpcGasSystemDaemon.cc	89
ftpcGasSystemSender.cc	90
ftpcGasSystemSender.hh	91
ftpcGasSystemSender_i.cc	92
ftpcGGridDaemon.cc	93
ftpcGGridSender.cc	94
ftpcGGridSender.hh	95
ftpcGGridSender_i.cc	96
ftpcHDLCTempsDaemon.cc	97
ftpcHDLCTempsSender.cc	98
ftpcHDLCTempsSender.hh	99
ftpcHDLCTempsSender_i.cc	100
ftpcHDLCVoltagesDaemon.cc	101
ftpcHDLCVoltagesSender.cc	102
ftpcHDLCVoltagesSender.hh	103
ftpcHDLCVoltagesSender_i.cc	104
ftpcTempsDaemon.cc	105
ftpcTempsSender.cc	106
ftpcTempsSender.hh	107
ftpcTempsSender_i.cc	108

Chapter 3

Doxxygen_MP_D_STAR_ONLINE_DB Class Documentation

3.1 fpcAnodesSender Class Reference

```
#include <fpcAnodesSender.hh>
```

Public Member Functions

- `fpcAnodesSender` (const char *localDir)
- virtual `~fpcAnodesSender` ()
- virtual void `initTable` ()
- virtual void `initTags` ()
- virtual void `initDataBase` ()
- virtual bool `loadUserControls` (const char *name, const char *value)
- virtual void `initQuery` ()
- virtual bool `queryData` ()
- virtual bool `readData` (const char *fileName)
- virtual bool `updateDb` (const char *fileName)
- virtual bool `readData` (ifstream &from)
- virtual bool `hasChanged` (int rowNumber)
- `char * readAny` ()
- `bool readVal` (`char *&value`)
- `bool readVal` (`float &value`)
- `bool readVal` (`double &value`)
- `bool readVal` (`short &value`)
- `bool readVal` (`int &value`)
- `bool readVal` (`long &value`)
- `bool readVal` (`long long &value`)
- `bool nextLine` (ifstream &from)
- `void readError` (int l, `char *c`, `char *m`)

Protected Attributes

- `ftpcAnodes previousVals [NUM_DB_ROWS]`
- `ftpcAnodes tempVals [NUM_DB_ROWS]`
- `int elementList [NUM_DB_ROWS]`
- `ftpcAnodes updateVals [NUM_DB_ROWS]`
- `int updateElements [NUM_DB_ROWS]`
- `bool mreadStatus`
- `char mline [256]`
- `char tmpline [256]`
- `char * ptr1`
- `char * ptr2`
- `float vdriftLimit`
dito
- `float cdriftLimit`

3.1.1 Constructor & Destructor Documentation

3.1.1.1 `ftpcAnodesSender::ftpcAnodesSender (const char * localDir)`

Definition at line 19 of file `ftpcAnodesSender.cc`.

```
19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("ftpcAnodes"); // setup the file I/O
24     initDataBase(); // database connections
25     initTable(); // table definitions
26
27 }
```

3.1.1.2 `virtual ftpcAnodesSender::~ftpcAnodesSender () [inline, virtual]`

Definition at line 42 of file `ftpcAnodesSender.hh`.

```
42 {};
```

3.1.2 Member Function Documentation

3.1.2.1 `bool ftpcAnodesSender::hasChanged (int rowNumber) [virtual]`

Definition at line 121 of file `ftpcAnodesSender_i.cc`.

```
121
122
123     ftpcAnodes* pre=&previousVals[rowNumber];
124     ftpcAnodes* cur=&tempVals[rowNumber];
125
126     if(fabs(pre->voltage1-cur->voltage1)>=vdriftLimit) return true;
127     if(fabs(pre->voltage2-cur->voltage2)>=vdriftLimit) return true;
```

```

128     if(fabs(pre->voltage3-cur->voltage3)>=vdriftLimit) return true;
129     if(fabs(pre->voltage4-cur->voltage4)>=vdriftLimit) return true;
130     if(fabs(pre->voltage5-cur->voltage5)>=vdriftLimit) return true;
131     if(fabs(pre->voltage6-cur->voltage6)>=vdriftLimit) return true;
132     if(fabs(pre->current1-cur->current1)>=cdriftLimit) return true;
133     if(fabs(pre->current2-cur->current2)>=cdriftLimit) return true;
134     if(fabs(pre->current3-cur->current3)>=cdriftLimit) return true;
135     if(fabs(pre->current4-cur->current4)>=cdriftLimit) return true;
136     if(fabs(pre->current5-cur->current5)>=cdriftLimit) return true;
137     if(fabs(pre->current6-cur->current6)>=cdriftLimit) return true;
138
139
140 /* example ... note -> change to any element requires db-update
141 * and thus returns true immediately
142 *
143 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
144 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
145 *
146 * ...
147 */
148
149 return false;
150 }
```

3.1.2.2 void ftpcAnodesSender::initDataBase () [virtual]

Definition at line 75 of file ftpcAnodesSender.cc.

```

75
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType   dbT = dbConditions;
81 StDbDomain dbD = dbFtpc;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.1.2.3 void ftpcAnodesSender::initQuery () [virtual]

Definition at line 44 of file ftpcAnodesSender_i.cc.

```

44
45 #define __METHOD__ "initQuery()"
46
47     ofstream to(queryFile);
48
49     if(!to.is_open()){
50         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
51         return;
52     }
53
54     const char* b[6]={"E","F","G","H","I","J"};
55     int j;
56     for(int i=0;i<2;i++){
57         for(j=0;j<6;j++)
58             to<<"ftpchv_rd_mv_0_"<<i<<"_"<<b[j]<<endl;
```

```

59         for(j=0;j<6;j++)
60             to<<"ftpchv_rd_mc_0_"<<i<<"_"<<b[j]<<endl;
61     }
62
63 /* example
64 *      for(int i=0;i<16;i++){
65 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
66 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
67 *          ....
68 *
69 *      }
70 */
71     to.close();
72
73 #undef __METHOD__
74 }

```

3.1.2.4 void ftpcAnodesSender::initTable () [virtual]

Definition at line 30 of file ftpcAnodesSender.cc.

```

30 {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("ftpcAnodes")))
35         sendMess("Could not find table=ftpcAnodes",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcAnodes));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(ftpcAnodes));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         ftpcAnodes* thv = (ftpcAnodes*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(ftpcAnodes));
55     }
56
57 #undef __METHOD__
58 };

```

3.1.2.5 void ftpcAnodesSender::initTags () [virtual]

Definition at line 66 of file ftpcAnodesSender.cc.

```

66 {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("ftpc");
70
71 }

```

3.1.2.6 bool ftpcAnodesSender::loadUserControls (const char * *name*, const char * *value*) [virtual]

Definition at line 20 of file ftpcAnodesSender_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24  * and duplicate this structure for each selection criteria
25 */
26 if(strcmp(name,"vdriftLimit")){
27     vdriftLimit=atof(value);
28     sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
29     return true;
30 }
31 if(strcmp(name,"cdriftLimit")){
32     cdriftLimit=atof(value);
33     sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;
35 }
36
37
38 return false;
39 #undef __METHOD__
40 }
```

3.1.2.7 bool ftpcAnodesSender::nextLine (ifstream & *from*) [inline]

Definition at line 75 of file ftpcAnodesSender.hh.

```

75
76     if(!from.getline(mline,255))return false;
77     return true;
78 }
```

3.1.2.8 bool ftpcAnodesSender::queryData () [virtual]

Definition at line 91 of file ftpcAnodesSender.cc.

```

91
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100 writeTime = (unsigned int)time(NULL);           //for database write time
101
102 //char systemCmd[1024];
103 ostringstream scmd;
104 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
105
106 if(system((scmd.str()).c_str()))
107     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
108
109
```

```

110     return true;
111 #undef __METHOD__
112 };

```

3.1.2.9 char * ftpcAnodesSender::readAny ()

Definition at line 197 of file ftpcAnodesSender.cc.

```

197
198
199     strcpy(tmpline,mline);
200     ptr1=tmpline;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;
203     ptr2=strtok(NULL," ");
204     return ptr2;
205 }

```

3.1.2.10 bool ftpcAnodesSender::readData (ifstream & from) [virtual]

Definition at line 80 of file ftpcAnodesSender_i.cc.

```

80
81 #define __METHOD__ "readData(ifstream)"
82
83 mreadStatus=true;
84 memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcAnodes));
85
86
87 int i=0;
88 char* c=__CLASS__;
89 char* m=__METHOD__;
90
91 for(i=0;i<NUM_DB_ROWS;i++){
92     if(!nextLine(from) || !readVal(tempVals[i].voltage1)) readError(__LINE__,c,m);
93     if(!nextLine(from) || !readVal(tempVals[i].voltage2)) readError(__LINE__,c,m);
94     if(!nextLine(from) || !readVal(tempVals[i].voltage3)) readError(__LINE__,c,m);
95     if(!nextLine(from) || !readVal(tempVals[i].voltage4)) readError(__LINE__,c,m);
96     if(!nextLine(from) || !readVal(tempVals[i].voltage5)) readError(__LINE__,c,m);
97     if(!nextLine(from) || !readVal(tempVals[i].voltage6)) readError(__LINE__,c,m);
98     if(!nextLine(from) || !readVal(tempVals[i].current1)) readError(__LINE__,c,m);
99     if(!nextLine(from) || !readVal(tempVals[i].current2)) readError(__LINE__,c,m);
100    if(!nextLine(from) || !readVal(tempVals[i].current3)) readError(__LINE__,c,m);
101    if(!nextLine(from) || !readVal(tempVals[i].current4)) readError(__LINE__,c,m);
102    if(!nextLine(from) || !readVal(tempVals[i].current5)) readError(__LINE__,c,m);
103    if(!nextLine(from) || !readVal(tempVals[i].current6)) readError(__LINE__,c,m);
104 }
105 /* example format
106  *   for(int i=0;i<NUM_DB_ROWS;i++){
107  *     if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
108  *
109  *     ....
110  *
111  *   }
112 */
113
114     from.close();
115 return true;
116 #undef __METHOD__
117 }

```

3.1.2.11 bool ftpcAnodesSender::readData (const char **fileName*) [virtual]

Definition at line 116 of file ftpcAnodesSender.cc.

```

116
117 #define __METHOD__ "readData(fileName)"
118
119   ifstream from(fileName);
120   if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122   return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.1.2.12 void ftpcAnodesSender::readError (int *l*, char **c*, char **m*) [inline]

Definition at line 80 of file ftpcAnodesSender.hh.

```

80
81   mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
82 }
```

3.1.2.13 bool ftpcAnodesSender::readVal (long long &*value*)

Definition at line 269 of file ftpcAnodesSender.cc.

```

269
270
271   if(!readAny())return false;
272   char* store[256];
273   value=strtoll(ptr2,store,10);
274   if(strlen(*store)>0) return false; // value is not a number
275
276   return true;
277 };
```

3.1.2.14 bool ftpcAnodesSender::readVal (long &*value*)

Definition at line 258 of file ftpcAnodesSender.cc.

```

258
259
260   if(!readAny())return false;
261
262   char* store[256];
263   value=strtol(ptr2,store,10);
264   if(strlen(*store)>0) return false; // value is not a number
265
266   return true;
267 };
```

3.1.2.15 bool ftcpAnodesSender::readVal (int & value)

Definition at line 247 of file ftcpAnodesSender.cc.

```
247 {  
248     if(!readAny()) return false;  
249  
250     char* store[256];  
251     value=(int)strtol(ptr2,store,10);  
252     if(strlen(*store)>0) return false; // value is not a number  
253  
254     return true;  
255 };
```

3.1.2.16 bool ftcpAnodesSender::readVal (short & value)

Definition at line 236 of file ftcpAnodesSender.cc.

```
236 {  
237     if(!readAny()) return false;  
238  
239     char* store[256];  
240     value=(short)strtol(ptr2,store,10);  
241     if(strlen(*store)>0) return false; // value is not a number  
242  
243     return true;  
244 };
```

3.1.2.17 bool ftcpAnodesSender::readVal (double & value)

Definition at line 225 of file ftcpAnodesSender.cc.

```
225 {  
226     if(!readAny())return false;  
227  
228     char* store[256];  
229     value=strtod(ptr2,store);  
230     if(strlen(*store)>0) return false; // value is not a number  
231  
232     return true;  
233 };
```

3.1.2.18 bool ftcpAnodesSender::readVal (float & value)

Definition at line 214 of file ftcpAnodesSender.cc.

```
214 {  
215     if(!readAny()) return false;  
216  
217     char* store[256];  
218     value=(float)strtod(ptr2,store);  
219     if(strlen(*store)>0) return false; // value is not a number  
220  
221     return true;  
222 };
```

3.1.2.19 bool ftpcAnodesSender::readVal (char *& value)

Definition at line 207 of file ftpcAnodesSender.cc.

```
207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

3.1.2.20 bool ftpcAnodesSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file ftpcAnodesSender.cc.

```
127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     ftpcAnodes* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159     //char mess[256];
160     ostringstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("ftpcAnodes");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176 #undef __METHOD__
177 }
```

3.1.3 Member Data Documentation

3.1.3.1 float **ftpcAnodesSender::cdriftLimit** [protected]

Definition at line 36 of file ftpcAnodesSender.hh.

3.1.3.2 int **ftpcAnodesSender::elementList[NUM_DB_ROWS]** [protected]

Definition at line 24 of file ftpcAnodesSender.hh.

3.1.3.3 char **ftpcAnodesSender::mline[256]** [protected]

Definition at line 29 of file ftpcAnodesSender.hh.

3.1.3.4 bool **ftpcAnodesSender::mreadStatus** [protected]

Definition at line 28 of file ftpcAnodesSender.hh.

3.1.3.5 ftpcAnodes **ftpcAnodesSender::previousVals[NUM_DB_ROWS]** [protected]

Definition at line 22 of file ftpcAnodesSender.hh.

3.1.3.6 char* **ftpcAnodesSender::ptr1** [protected]

Definition at line 31 of file ftpcAnodesSender.hh.

3.1.3.7 char * **ftpcAnodesSender::ptr2** [protected]

Definition at line 31 of file ftpcAnodesSender.hh.

3.1.3.8 ftpcAnodes **ftpcAnodesSender::tempVals[NUM_DB_ROWS]** [protected]

Definition at line 23 of file ftpcAnodesSender.hh.

3.1.3.9 char **ftpcAnodesSender::tmpline[256]** [protected]

Definition at line 30 of file ftpcAnodesSender.hh.

3.1.3.10 int **ftpcAnodesSender::updateElements[NUM_DB_ROWS]** [protected]

Definition at line 26 of file ftpcAnodesSender.hh.

3.1.3.11 `ftpcAnodes` `ftpcAnodesSender::updateVals[NUM_DB_ROWS]` [protected]

Definition at line 25 of file `ftpcAnodesSender.hh`.

3.1.3.12 `float` `ftpcAnodesSender::vdriftLimit` [protected]

dito

Definition at line 35 of file `ftpcAnodesSender.hh`.

The documentation for this class was generated from the following files:

- `ftpcAnodesSender.hh`
- `ftpcAnodesSender.cc`
- `ftpcAnodesSender_i.cc`

3.2 ftpcCathodeSender Class Reference

```
#include <ftpcCathodeSender.hh>
```

Public Member Functions

- `ftpcCathodeSender` (const char *localDir)
- virtual `~ftpcCathodeSender` ()
- virtual void `initTable` ()
- virtual void `initTags` ()
- virtual void `initDataBase` ()
- virtual bool `loadUserControls` (const char *name, const char *value)
- virtual void `initQuery` ()
- virtual bool `queryData` ()
- virtual bool `readData` (const char *fileName)
- virtual bool `updateDb` (const char *fileName)
- virtual bool `readData` (ifstream &from)
- virtual bool `hasChanged` (int rowNumber)
- char * `readAny` ()
- bool `readVal` (char *&value)
- bool `readVal` (float &value)
- bool `readVal` (double &value)
- bool `readVal` (short &value)
- bool `readVal` (int &value)
- bool `readVal` (long &value)
- bool `readVal` (long long &value)
- bool `nextLine` (ifstream &from)
- void `readError` (int l, char *c, char *m)

Protected Attributes

- `ftpcCathode previousVals [NUM_DB_ROWS]`
 - `ftpcCathode tempVals [NUM_DB_ROWS]`
 - int `elementList [NUM_DB_ROWS]`
 - `ftpcCathode updateVals [NUM_DB_ROWS]`
 - int `updateElements [NUM_DB_ROWS]`
 - bool `mreadStatus`
 - char `mline [256]`
 - char `tmpline [256]`
 - char * `ptr1`
 - char * `ptr2`
 - float `vdriftLimit`
- dito*
- float `cdriftLimit`

3.2.1 Constructor & Destructor Documentation

3.2.1.1 ftpcCathodeSender::ftpcCathodeSender (const char * *localDir*)

Definition at line 19 of file ftpcCathodeSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("ftpcCathode"); // setup the file I/O
24     initDataBase();      // database connections
25     initTable();         // table definitions
26
27 }
```

3.2.1.2 virtual ftpcCathodeSender::~ftpcCathodeSender () [inline, virtual]

Definition at line 43 of file ftpcCathodeSender.hh.

```
43 {};
```

3.2.2 Member Function Documentation

3.2.2.1 bool ftpcCathodeSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 108 of file ftpcCathodeSender_i.cc.

```

108
109
110 ftpcCathode* pre=&previousVals[rowNumber];
111 ftpcCathode* cur=&tempVals[rowNumber];
112
113 if(fabs(pre->voltage-cur->voltage)>=vdriftLimit) return true;
114 if(fabs(pre->current-cur->current)>=cdriftLimit) return true;
115
116 /* example ... note -> change to any element requires db-update
117 * and thus returns true immediately
118 *
119 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
120 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
121 *
122 * ...
123 */
124
125 return false;
126 }
```

3.2.2.2 void ftpcCathodeSender::initDataBase () [virtual]

Definition at line 75 of file ftpcCathodeSender.cc.

```

75
76 #define __METHOD__ "initDataBase()"
77 }
```

```

78     /* More than an example... swap user & dbTrg as per subsystem*/
79     mgr->setUser("stardb","");
80     StDbType    dbT = dbConditions;
81     StDbDomain dbD = dbFtpc;
82
83     if( !( node = mgr->initConfig(dbT,dbD)) )
84         sendMess( "Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.2.2.3 void ftpcCathodeSender::initQuery () [virtual]

Definition at line 43 of file ftpcCathodeSender_i.cc.

```

43 {
44 #define __METHOD__ "initQuery()"
45
46     ofstream to(queryFile);
47
48     if(!to.is_open()){
49         sendMess( "Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
50         return;
51     }
52
53     to<<"ftpc_mv_w" << endl;
54     to<<"ftpc_mi_w" << endl;
55     to<<"ftpc_mv_e" << endl;
56     to<<"ftpc_mi_e" << endl;
57
58     /* example
59      *      for(int i=0;i<16;i++){
60      *          to<<"TRGhv:SUB_RD_V_1:" << i << ".E" << endl;
61      *          to<<"TRGhv:SUB_RD_V_1:" << i << ".F" << endl;
62      *          ....
63      *      }
64      */
65
66     to.close();
67
68 #undef __METHOD__
69
70 }
```

3.2.2.4 void ftpcCathodeSender::initTable () [virtual]

Definition at line 30 of file ftpcCathodeSender.cc.

```

30     {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("ftpcCathode")))
35         sendMess( "Could not find table=ftpcCathode",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcCathode));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(ftpcCathode));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
```

```

42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         ftpcCathode* thv = (ftpcCathode*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(ftpcCathode));
55     }
56
57 #undef __METHOD__
58 };

```

3.2.2.5 void ftpcCathodeSender::initTags () [virtual]

Definition at line 66 of file ftpcCathodeSender.cc.

```

66 {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("ftpc");
70
71 }

```

3.2.2.6 bool ftpcCathodeSender::loadUserControls (const char * *name*, const char * *value*) [virtual]

Definition at line 20 of file ftpcCathodeSender_i.cc.

```

20 {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 */
26 if strstr(name,"vdriftLimit")){
27     vdriftLimit=atof(value);
28     sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
29     return true;
30 }
31 if strstr(name,"cdriftLimit")){
32     cdriftLimit=atof(value);
33     sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;
35 }
36
37 return false;
38 #undef __METHOD__
39 }

```

3.2.2.7 bool ftpcCathodeSender::nextLine (ifstream & *from*) [inline]

Definition at line 76 of file ftpcCathodeSender.hh.

```

76
77     if(!from.getline(mline,255))return false;
78     return true;
79 }
```

3.2.2.8 bool ftpcCathodeSender::queryData () [virtual]

Definition at line 91 of file ftpcCathodeSender.cc.

```

91
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101    writeTime = (unsigned int)time(NULL);           //for database write time
102
103    //char systemCmd[1024];
104    ostringstream scmd;
105    scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107    if(system((scmd.str()).c_str()))
108        return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110    return true;
111 #undef __METHOD__
112 };
```

3.2.2.9 char * ftpcCathodeSender::readAny ()

Definition at line 197 of file ftpcCathodeSender.cc.

```

197
198
199    strcpy(tmpLine,mline);
200    ptr1=tmpLine;
201    ptr2=strtok(ptr1," ");
202    if(!ptr2) return ptr2;
203    ptr2=strtok(NULL," ");
204    return ptr2;
205 }
```

3.2.2.10 bool ftpcCathodeSender::readData (ifstream &*from*) [virtual]

Definition at line 76 of file ftpcCathodeSender_i.cc.

```

76
77 #define __METHOD__ "readData(ifstream)"
78
79 mreadStatus=true;
80 memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcCathode));
81
82 }
```

```

83 int i=0;
84 char* c=__CLASS__;
85 char* m=__METHOD__;
86
87 for(i=0;i<NUM_DB_ROWS;i++){
88     if(!nextLine(from) || !readVal(tempVals[i].voltage)) readError(__LINE__,c,m);
89     if(!nextLine(from) || !readVal(tempVals[i].current)) readError(__LINE__,c,m);
90 }
91
92 /* example format
93 *   for(int i=0;i<NUM_DB_ROWS;i++){
94 *     if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
95 *
96 *     ....
97 *
98 *   }
99 */
100
101 from.close();
102 return true;
103 #undef __METHOD__
104 }
```

3.2.2.11 bool ftpcCathodeSender::readData (const char **fileName*) [virtual]

Definition at line 116 of file ftpcCathodeSender.cc.

```

116 {
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.2.2.12 void ftpcCathodeSender::readError (int *l*, char **c*, char **m*) [inline]

Definition at line 81 of file ftpcCathodeSender.hh.

```

81 {
82     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
83 }
```

3.2.2.13 bool ftpcCathodeSender::readVal (long long &*value*)

Definition at line 269 of file ftpcCathodeSender.cc.

```

269 {
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

3.2.2.14 bool ftpcCathodeSender::readVal (long & value)

Definition at line 258 of file ftpcCathodeSender.cc.

```
258 {  
259     if(!readAny())return false;  
260  
261     char* store[256];  
262     value=strtol(ptr2,store,10);  
263     if(strlen(*store)>0) return false; // value is not a number  
264  
265     return true;  
266 };
```

3.2.2.15 bool ftpcCathodeSender::readVal (int & value)

Definition at line 247 of file ftpcCathodeSender.cc.

```
247 {  
248     if(!readAny()) return false;  
249  
250     char* store[256];  
251     value=(int)strtol(ptr2,store,10);  
252     if(strlen(*store)>0) return false; // value is not a number  
253  
254     return true;  
255 };
```

3.2.2.16 bool ftpcCathodeSender::readVal (short & value)

Definition at line 236 of file ftpcCathodeSender.cc.

```
236 {  
237     if(!readAny()) return false;  
238  
239     char* store[256];  
240     value=(short)strtol(ptr2,store,10);  
241     if(strlen(*store)>0) return false; // value is not a number  
242  
243     return true;  
244 };
```

3.2.2.17 bool ftpcCathodeSender::readVal (double & value)

Definition at line 225 of file ftpcCathodeSender.cc.

```
225 {  
226     if(!readAny())return false;  
227  
228     char* store[256];  
229     value=strtod(ptr2,store);  
230     if(strlen(*store)>0) return false; // value is not a number  
231  
232     return true;  
233 };
```

3.2.2.18 bool ftpcCathodeSender::readVal (float & value)

Definition at line 214 of file ftpcCathodeSender.cc.

```

214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };

```

3.2.2.19 bool ftpcCathodeSender::readVal (char *& value)

Definition at line 207 of file ftpcCathodeSender.cc.

```

207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }

```

3.2.2.20 bool ftpcCathodeSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file ftpcCathodeSender.cc.

```

127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     ftpcCathode* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }

```

```

156
157     if(numRows==0) return sendMess(" No update required for ",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159     //char mess[256];
160     ostringstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("ftpcCathode");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175     return true;
176 #undef __METHOD__
177 }
```

3.2.3 Member Data Documentation

3.2.3.1 float **ftpcCathodeSender::cdriftLimit** [protected]

Definition at line 37 of file ftpcCathodeSender.hh.

3.2.3.2 int **ftpcCathodeSender::elementList[NUM_DB_ROWS]** [protected]

Definition at line 24 of file ftpcCathodeSender.hh.

3.2.3.3 char **ftpcCathodeSender::mline[256]** [protected]

Definition at line 29 of file ftpcCathodeSender.hh.

3.2.3.4 bool **ftpcCathodeSender::mreadStatus** [protected]

Definition at line 28 of file ftpcCathodeSender.hh.

3.2.3.5 ftpcCathode **ftpcCathodeSender::previousVals[NUM_DB_ROWS]** [protected]

Definition at line 22 of file ftpcCathodeSender.hh.

3.2.3.6 char* **ftpcCathodeSender::ptr1** [protected]

Definition at line 31 of file ftpcCathodeSender.hh.

3.2.3.7 char * **ftpcCathodeSender::ptr2** [protected]

Definition at line 31 of file ftpcCathodeSender.hh.

3.2.3.8 ftpcCathode **ftpcCathodeSender::tempVals[NUM_DB_ROWS] [protected]**

Definition at line 23 of file ftpcCathodeSender.hh.

3.2.3.9 char **ftpcCathodeSender::tmpLine[256] [protected]**

Definition at line 30 of file ftpcCathodeSender.hh.

3.2.3.10 int **ftpcCathodeSender::updateElements[NUM_DB_ROWS] [protected]**

Definition at line 26 of file ftpcCathodeSender.hh.

3.2.3.11 ftpcCathode **ftpcCathodeSender::updateVals[NUM_DB_ROWS] [protected]**

Definition at line 25 of file ftpcCathodeSender.hh.

3.2.3.12 float **ftpcCathodeSender::vdriftLimit [protected]**

dito

Definition at line 36 of file ftpcCathodeSender.hh.

The documentation for this class was generated from the following files:

- **ftpcCathodeSender.hh**
- **ftpcCathodeSender.cc**
- **ftpcCathodeSender_i.cc**

3.3 ftpcGasSystemSender Class Reference

```
#include <ftpcGasSystemSender.hh>
```

Public Member Functions

- [ftpcGasSystemSender](#) (const char *localDir)
- [virtual ~ftpcGasSystemSender](#) ()
- [virtual void initTable](#) ()
- [virtual void initTags](#) ()
- [virtual void initDataBase](#) ()
- [virtual bool loadUserControls](#) (const char *name, const char *value)
- [virtual void initQuery](#) ()
- [virtual bool queryData](#) ()
- [virtual bool readData](#) (const char *fileName)
- [virtual bool updateDb](#) (const char *fileName)
- [virtual bool readData](#) (ifstream &from)
- [virtual bool hasChanged](#) (int rowNumber)
- [char * readAny](#) ()
- [bool readVal](#) (char *&value)
- [bool readVal](#) (float &value)
- [bool readVal](#) (double &value)
- [bool readVal](#) (short &value)
- [bool readVal](#) (int &value)
- [bool readVal](#) (long &value)
- [bool readVal](#) (long long &value)
- [bool nextLine](#) (ifstream &from)
- [void readError](#) (int l, char *c, char *m)

Protected Attributes

- [ftpcGasSystem previousVals](#) [NUM_DB_ROWS]
- [ftpcGasSystem tempVals](#) [NUM_DB_ROWS]
- [int elementList](#) [NUM_DB_ROWS]
- [ftpcGasSystem updateVals](#) [NUM_DB_ROWS]
- [int updateElements](#) [NUM_DB_ROWS]
- [bool mreadStatus](#)
- [char mline](#) [256]
- [char tmpline](#) [256]
- [char * ptr1](#)
- [char * ptr2](#)
- [float o2driftLimit](#)
dito
- [float h2odriftLimit](#)
- [float flowdriftLimit](#)

3.3.1 Constructor & Destructor Documentation

3.3.1.1 ft pcGasSystemSender::ft pcGasSystemSender (const char * *localDir*)

Definition at line 19 of file ft pcGasSystemSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("ft pcGasSystem"); // setup the file I/O
24     init DataBase(); // database connections
25     init Table(); // table definitions
26
27 }
```

3.3.1.2 virtual ft pcGasSystemSender::~ft pcGasSystemSender () [inline, virtual]

Definition at line 44 of file ft pcGasSystemSender.hh.

```
44 {};
```

3.3.2 Member Function Documentation

3.3.2.1 bool ft pcGasSystemSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 132 of file ft pcGasSystemSender_i.cc.

```

132
133
134 ft pcGasSystem* pre=&previousVals[rowNumber];
135 ft pcGasSystem* cur=&tempVals[rowNumber];
136
137 if(fabs(pre->westO2ppm-cur->westO2ppm)>=o2driftLimit) return true;
138 if(fabs(pre->eastO2ppm-cur->eastO2ppm)>=o2driftLimit) return true;
139 if(fabs(pre->extO2ppm-cur->extO2ppm)>=o2driftLimit) return true;
140 if(fabs(pre->westH2Odp-cur->westH2Odp)>=h2odriftLimit) return true;
141 if(fabs(pre->eastH2Odp-cur->eastH2Odp)>=h2odriftLimit) return true;
142 if(fabs(pre->ArFlow-cur->ArFlow)>=flowdriftLimit) return true;
143 if(fabs(pre->CO2Flow-cur->CO2Flow)>=flowdriftLimit) return true;
144 if(strcmp(pre->status,cur->status) != 0) return true;
145
146 /* example ... note -> change to any element requires db-update
147 * and thus returns true immediately
148 *
149 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
150 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
151 *
152 * ...
153 */
154
155 return false;
156 }
```

3.3.2.2 void ft pcGasSystemSender::init DataBase () [virtual]

Definition at line 75 of file ft pcGasSystemSender.cc.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
77
78     /* More than an example... swap user & dbTrg as per subsystem*/
79     mgr->setUser("stardb","");
80     StDbType    dbT = dbConditions;
81     StDbDomain dbD = dbFtpc;
82
83     if( !( node = mgr->initConfig(dbT,dbD)) )
84         sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.3.2.3 void ftpcGasSystemSender::initQuery () [virtual]

Definition at line 49 of file ftpcGasSystemSender_i.cc.

```

49                                     {
50 #define __METHOD__ "initQuery()"
51
52     ofstream to(queryFile);
53
54     if(!to.is_open()){
55         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
56         return;
57     }
58
59     to<<"ftpc_gas_o2_w_ppm"<<endl;
60     to<<"ftpc_gas_o2_w_mv"<<endl;
61     to<<"ftpc_gas_o2_e_ppm"<<endl;
62     to<<"ftpc_gas_o2_e_mv"<<endl;
63     to<<"ftpc_gas_o2_ext_ppm"<<endl;
64     to<<"ftpc_gas_o2_ext_mv"<<endl;
65     to<<"ftpc_gas_h2o_w_dp"<<endl;
66     to<<"ftpc_gas_h2o_e_dp"<<endl;
67     to<<"ftpc_gas_ar_flow_lh"<<endl;
68     to<<"ftpc_gas_co2_flow_lh"<<endl;
69     to<<"ftpc_gas_status_out"<<endl;
70
71 /* example
72 *      for(int i=0;i<16;i++){
73 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.E"<<endl;
74 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.F"<<endl;
75 *          ....
76 *
77 *      }
78 */
79     to.close();
80
81 #undef __METHOD__
82 }
```

3.3.2.4 void ftpcGasSystemSender::initTable () [virtual]

Definition at line 30 of file ftpcGasSystemSender.cc.

```

30                                     {
31 #define __METHOD__ "initTable()"
32
```

```

33     StDbTable* table=0;
34     if(!(table=node->addDbTable("ftpcGasSystem")))
35         sendMess("Could not find table=ftpcGasSystem",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcGasSystem));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(ftpcGasSystem));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         ftpcGasSystem* thv = (ftpcGasSystem*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(ftpcGasSystem));
55     }
56
57 #undef __METHOD__
58 };

```

3.3.2.5 void ftpcGasSystemSender::initTags () [virtual]

Definition at line 66 of file ftpcGasSystemSender.cc.

```

66 {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("ftpc");
70 }
71 }

```

3.3.2.6 bool ftpcGasSystemSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file ftpcGasSystemSender_i.cc.

```

20 {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24  * and duplicate this structure for each selection criteria
25 */
26
27 if(strstr(name,"o2driftLimit")){
28     o2driftLimit=atof(value);
29     sendMess("o2driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
30     return true;
31 }
32 if(strstr(name,"h2odriftLimit")){
33     h2odriftLimit=atof(value);
34     sendMess("h2odriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
35     return true;
36 }

```

```

37 if(strstr(name,"flowdriftLimit")){
38   flowdriftLimit=atof(value);
39   sendMess("flowdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
40   return true;
41 }
42
43 return false;
44 #undef __METHOD__
45 }
```

3.3.2.7 bool **ftpcGasSystemSender::nextLine (ifstream & from) [inline]**

Definition at line 77 of file ftpcGasSystemSender.hh.

```

77
78   if(!from.getline(mline,255))return false;
79   return true;
80 }
```

3.3.2.8 bool **ftpcGasSystemSender::queryData () [virtual]**

Definition at line 91 of file ftpcGasSystemSender.cc.

```

91
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108   return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

3.3.2.9 char * **ftpcGasSystemSender::readAny ()**

Definition at line 197 of file ftpcGasSystemSender.cc.

```

197
198
199 strcpy(tmpLine,mline);
200 ptr1=tmpLine;
201 ptr2=strtok(ptr1," ");
202 if(!ptr2) return ptr2;
203 ptr2=strtok(NULL," ");
204 return ptr2;
205 }
```

3.3.2.10 bool ftpcGasSystemSender::readData (ifstream &from) [virtual]

Definition at line 88 of file ftpcGasSystemSender_i.cc.

```

88
89 #define __METHOD__ "readData(ifstream)"
90
91 mreadStatus=true;
92 memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcGasSystem));
93
94
95 int i=0;
96 char* c=__CLASS__;
97 char* m=__METHOD__;
98
99 if(!nextLine(from) || !readVal(tempVals[0].westO2ppm)) readError(__LINE__,c,m);
100 if(!nextLine(from) || !readVal(tempVals[0].westO2mv)) readError(__LINE__,c,m);
101 if(!nextLine(from) || !readVal(tempVals[0].eastO2ppm)) readError(__LINE__,c,m);
102 if(!nextLine(from) || !readVal(tempVals[0].eastO2mv)) readError(__LINE__,c,m);
103 if(!nextLine(from) || !readVal(tempVals[0].extO2ppm)) readError(__LINE__,c,m);
104 if(!nextLine(from) || !readVal(tempVals[0].extO2mv)) readError(__LINE__,c,m);
105 if(!nextLine(from) || !readVal(tempVals[0].westH2Odp)) readError(__LINE__,c,m);
106 if(!nextLine(from) || !readVal(tempVals[0].eastH2Odp)) readError(__LINE__,c,m);
107 if(!nextLine(from) || !readVal(tempVals[0].ArFlow)) readError(__LINE__,c,m);
108 if(!nextLine(from) || !readVal(tempVals[0].CO2Flow)) readError(__LINE__,c,m);
109 //MPD Changes
110     char* tmpval0;
111     if(!nextLine(from)){
112         tmpval0=tempVals[i].status;
113         if(!readVal(tmpval0))readError(__LINE__,__CLASS__,__METHOD__);
114     }
115 //if(!nextLine(from) || !readVal(tempVals[0].status)) readError(__LINE__,c,m);
116 /* example format
117 *   for(int i=0;i<NUM_DB_ROWS;i++){
118 *     if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
119 *
120 *     ....
121 *
122 *   }
123 */
124
125     from.close();
126 return true;
127 #undef __METHOD__
128 }
```

3.3.2.11 bool ftpcGasSystemSender::readData (const char *fileName) [virtual]

Definition at line 116 of file ftpcGasSystemSender.cc.

```

116
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.3.2.12 void **ftpcGasSystemSender::readError** (int *l*, char * *c*, char * *m*) [inline]

Definition at line 82 of file ftpcGasSystemSender.hh.

```
82
83     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
84 }
```

3.3.2.13 bool **ftpcGasSystemSender::readVal** (long long & *value*)

Definition at line 269 of file ftpcGasSystemSender.cc.

```
269
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

3.3.2.14 bool **ftpcGasSystemSender::readVal** (long & *value*)

Definition at line 258 of file ftpcGasSystemSender.cc.

```
258
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

3.3.2.15 bool **ftpcGasSystemSender::readVal** (int & *value*)

Definition at line 247 of file ftpcGasSystemSender.cc.

```
247
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

3.3.2.16 bool ft pcGasSystemSender::readVal (short & *value*)

Definition at line 236 of file ft pcGasSystemSender.cc.

```
236
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

3.3.2.17 bool ft pcGasSystemSender::readVal (double & *value*)

Definition at line 225 of file ft pcGasSystemSender.cc.

```
225
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

3.3.2.18 bool ft pcGasSystemSender::readVal (float & *value*)

Definition at line 214 of file ft pcGasSystemSender.cc.

```
214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

3.3.2.19 bool ft pcGasSystemSender::readVal (char *& *value*)

Definition at line 207 of file ft pcGasSystemSender.cc.

```
207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

3.3.2.20 bool **ftpcGasSystemSender::updateDb (const char *fileName)** [virtual]

Definition at line 127 of file ftpcGasSystemSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     ftpcGasSystem* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159 //char mess[256];
160 ostringstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("ftpcGasSystem");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175     return true;
176 #undef __METHOD__
177 }
```

3.3.3 Member Data Documentation

3.3.3.1 int **ftpcGasSystemSender::elementList[NUM_DB_ROWS]** [protected]

Definition at line 24 of file ftpcGasSystemSender.hh.

3.3.3.2 float `ftpcGasSystemSender::flowdriftLimit` [protected]

Definition at line 38 of file ftpcGasSystemSender.hh.

3.3.3.3 float `ftpcGasSystemSender::h2odriftLimit` [protected]

Definition at line 37 of file ftpcGasSystemSender.hh.

3.3.3.4 char `ftpcGasSystemSender::mline[256]` [protected]

Definition at line 29 of file ftpcGasSystemSender.hh.

3.3.3.5 bool `ftpcGasSystemSender::mreadStatus` [protected]

Definition at line 28 of file ftpcGasSystemSender.hh.

3.3.3.6 float `ftpcGasSystemSender::o2driftLimit` [protected]

dito

Definition at line 36 of file ftpcGasSystemSender.hh.

3.3.3.7 `ftpcGasSystem` `ftpcGasSystemSender::previousVals[NUM_DB_ROWS]` [protected]

Definition at line 22 of file ftpcGasSystemSender.hh.

3.3.3.8 char* `ftpcGasSystemSender::ptr1` [protected]

Definition at line 31 of file ftpcGasSystemSender.hh.

3.3.3.9 char * `ftpcGasSystemSender::ptr2` [protected]

Definition at line 31 of file ftpcGasSystemSender.hh.

3.3.3.10 `ftpcGasSystem` `ftpcGasSystemSender::tempVals[NUM_DB_ROWS]` [protected]

Definition at line 23 of file ftpcGasSystemSender.hh.

3.3.3.11 char `ftpcGasSystemSender::tmpline[256]` [protected]

Definition at line 30 of file ftpcGasSystemSender.hh.

3.3.3.12 int `ftpcGasSystemSender::updateElements[NUM_DB_ROWS]` [protected]

Definition at line 26 of file ftpcGasSystemSender.hh.

3.3.3.13 **ftpcGasSystem [ftpcGasSystemSender::updateVals\[**NUM_DB_ROWS**\]](#)** [protected]

Definition at line 25 of file [ftpcGasSystemSender.hh](#).

The documentation for this class was generated from the following files:

- [ftpcGasSystemSender.hh](#)
- [ftpcGasSystemSender.cc](#)
- [ftpcGasSystemSender_i.cc](#)

3.4 **ftpcGGridSender** Class Reference

```
#include <ftpcGGridSender.hh>
```

Public Member Functions

- `ftpcGGridSender` (const char *localDir)
- virtual `~ftpcGGridSender` ()
- virtual void `initTable` ()
- virtual void `initTags` ()
- virtual void `initDataBase` ()
- virtual bool `loadUserControls` (const char *name, const char *value)
- virtual void `initQuery` ()
- virtual bool `queryData` ()
- virtual bool `readData` (const char *fileName)
- virtual bool `updateDb` (const char *fileName)
- virtual bool `readData` (ifstream &from)
- virtual bool `hasChanged` (int rowNumber)
- char * `readAny` ()
- bool `readVal` (char *&value)
- bool `readVal` (float &value)
- bool `readVal` (double &value)
- bool `readVal` (short &value)
- bool `readVal` (int &value)
- bool `readVal` (long &value)
- bool `readVal` (long long &value)
- bool `nextLine` (ifstream &from)
- void `readError` (int l, char *c, char *m)

Protected Attributes

- `ftpcGGrid previousVals [NUM_DB_ROWS]`
- `ftpcGGrid tempVals [NUM_DB_ROWS]`
- int `elementList [NUM_DB_ROWS]`
- `ftpcGGrid updateVals [NUM_DB_ROWS]`
- int `updateElements [NUM_DB_ROWS]`
- bool `mreadStatus`
- char `mline [256]`
- char `tmpline [256]`
- char * `ptr1`
- char * `ptr2`
- float `vdriftLimit`

dito

3.4.1 Constructor & Destructor Documentation

3.4.1.1 **ftpcGGridSender::ftpcGGridSender (const char * *localDir*)**

Definition at line 19 of file ftpcGGridSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("ftpcGGrid"); // setup the file I/O
24     initDataBase(); // database connections
25     initTable(); // table definitions
26
27 }
```

3.4.1.2 **virtual ftpcGGridSender::~ftpcGGridSender () [inline, virtual]**

Definition at line 42 of file ftpcGGridSender.hh.

```
42 {};
```

3.4.2 Member Function Documentation

3.4.2.1 **bool ftpcGGridSender::hasChanged (int *rowNumber*) [virtual]**

Definition at line 118 of file ftpcGGridSender_i.cc.

```

118
119
120 ftpcGGrid* pre=&previousVals[rowNumber];
121 ftpcGGrid* cur=&tempVals[rowNumber];
122
123 if(fabs(pre->hiVolt_s1-cur->hiVolt_s1)>=vdriftLimit) return true;
124 if(fabs(pre->loVolt_s1-cur->loVolt_s1)>=vdriftLimit) return true;
125 if(fabs(pre->openVolt_s1-cur->openVolt_s1)>=vdriftLimit) return true;
126
127 if(fabs(pre->hiVolt_s2-cur->hiVolt_s2)>=vdriftLimit) return true;
128 if(fabs(pre->loVolt_s2-cur->loVolt_s2)>=vdriftLimit) return true;
129 if(fabs(pre->openVolt_s2-cur->openVolt_s2)>=vdriftLimit) return true;
130
131 if(fabs(pre->hiVolt_s3-cur->hiVolt_s3)>=vdriftLimit) return true;
132 if(fabs(pre->loVolt_s3-cur->loVolt_s3)>=vdriftLimit) return true;
133 if(fabs(pre->openVolt_s3-cur->openVolt_s3)>=vdriftLimit) return true;
134
135 if(fabs(pre->hiVolt_s4-cur->hiVolt_s4)>=vdriftLimit) return true;
136 if(fabs(pre->loVolt_s4-cur->loVolt_s4)>=vdriftLimit) return true;
137 if(fabs(pre->openVolt_s4-cur->openVolt_s4)>=vdriftLimit) return true;
138
139 /* example ... note -> change to any element requires db-update
140  * and thus returns true immediately
141  */
142
143 /*if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
144 /*if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
145 */
146 /* ....
147 */
```

```
149 return false;
150 }
```

3.4.2.2 void ftpcGGridSender::initDataBase () [virtual]

Definition at line 75 of file ftpcGGridSender.cc.

```
75 {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbFtpc;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.4.2.3 void ftpcGGridSender::initQuery () [virtual]

Definition at line 39 of file ftpcGGridSender_i.cc.

```
39 {
40 #define __METHOD__ "initQuery()"
41
42     ofstream to(queryFile);
43
44     if(!to.is_open()){
45         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
46         return;
47     }
48
49     const char* abc[3]={"A","B","C"};
50     for(int j=1;j<9;j++)
51         for(int i=0;i<3;i++)to<<"FGG_A_In_"<<abc[i]<<"_<<j<<endl;
52
53
54 /* example
55 *      for(int i=0;i<16;i++){
56 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.E"<<endl;
57 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.F"<<endl;
58 *          ....
59 *
60 *      }
61 */
62
63     to.close();
64
65 #undef __METHOD__
66 }
```

3.4.2.4 void ftpcGGridSender::initTable () [virtual]

Definition at line 30 of file ftpcGGridSender.cc.

```

30
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("ftpcGGrid")))
35         sendMess("Could not find table=ftpcGGrid",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcGGrid));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(ftpcGGrid));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled(" <<NUM_DB_ROWS<< ")" <<ends;
46         sendMess(ms.str().c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         ftpcGGrid* thv = (ftpcGGrid*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(ftpcGGrid));
55     }
56
57 #undef __METHOD__
58 };

```

3.4.2.5 void **ftpcGGridSender::initTags ()** [virtual]

Definition at line 66 of file ftpcGGridSender.cc.

```

66
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("ftpc");
70
71 }

```

3.4.2.6 bool **ftpcGGridSender::loadUserControls (const char * name, const char * value)** [virtual]

Definition at line 20 of file ftpcGGridSender_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 */
26     if strstr(name,"vdriftLimit")){
27         vdriftLimit=atof(value);
28         sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
29         return true;
30     }
31
32
33 return false;
34 #undef __METHOD__
35 }

```

3.4.2.7 bool ft pcGGridSender::nextLine (ifstream & from) [inline]

Definition at line 75 of file ft pcGGridSender.hh.

```
75
76     if(!from.getline(mline,255))return false;
77     return true;
78 }
```

3.4.2.8 bool ft pcGGridSender::queryData () [virtual]

Definition at line 91 of file ft pcGGridSender.cc.

```
91
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

3.4.2.9 char * ft pcGGridSender::readAny ()

Definition at line 197 of file ft pcGGridSender.cc.

```
197
198
199 strcpy(tmpLine,mline);
200 ptr1=tmpLine;
201 ptr2=strtok(ptr1," ");
202 if(!ptr2) return ptr2;
203 ptr2=strtok(NULL," ");
204 return ptr2;
205 }
```

3.4.2.10 bool ft pcGGridSender::readData (ifstream & from) [virtual]

Definition at line 71 of file ft pcGGridSender_i.cc.

```
71
72 #define __METHOD__ "readData(ifstream)"
```

```

73
74 mreadStatus=true;
75 memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcGGrid));
76
77
78 int i=0;
79 char* c=__CLASS__;
80 char* m=__METHOD__;
81
82 for(i=0;i<NUM_DB_ROWS;i++){
83
84     if(!nextLine(from) || !readVal(tempVals[i].hiVolt_s1)) readError(__LINE__,c,m);
85     if(!nextLine(from) || !readVal(tempVals[i].loVolt_s1)) readError(__LINE__,c,m);
86     if(!nextLine(from) || !readVal(tempVals[i].openVolt_s1)) readError(__LINE__,c,m);
87
88     if(!nextLine(from) || !readVal(tempVals[i].hiVolt_s2)) readError(__LINE__,c,m);
89     if(!nextLine(from) || !readVal(tempVals[i].loVolt_s2)) readError(__LINE__,c,m);
90     if(!nextLine(from) || !readVal(tempVals[i].openVolt_s2)) readError(__LINE__,c,m);
91
92     if(!nextLine(from) || !readVal(tempVals[i].hiVolt_s3)) readError(__LINE__,c,m);
93     if(!nextLine(from) || !readVal(tempVals[i].loVolt_s3)) readError(__LINE__,c,m);
94     if(!nextLine(from) || !readVal(tempVals[i].openVolt_s3)) readError(__LINE__,c,m);
95
96     if(!nextLine(from) || !readVal(tempVals[i].hiVolt_s4)) readError(__LINE__,c,m);
97     if(!nextLine(from) || !readVal(tempVals[i].loVolt_s4)) readError(__LINE__,c,m);
98     if(!nextLine(from) || !readVal(tempVals[i].openVolt_s4)) readError(__LINE__,c,m);
99
100 }
101
102 /* example format
103 *  for(int i=0;i<NUM_DB_ROWS;i++){
104 *      if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
105 *
106 *      ....
107 *
108 *  }
109 */
110
111 from.close();
112 return true;
113 #undef __METHOD__
114 }
```

3.4.2.11 bool ftpcGGridSender::readData (const char * *fileName*) [virtual]

Definition at line 116 of file ftpcGGridSender.cc.

```

116                                         {
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.4.2.12 void ftpcGGridSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 80 of file ftpcGGridSender.hh.

```

80
81   mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
82 }
```

3.4.2.13 bool ftpcGGridSender::readVal (long long & value)

Definition at line 269 of file ftpcGGridSender.cc.

```

269
270
271   if(!readAny())return false;
272   char* store[256];
273   value=strtoll(ptr2,store,10);
274   if(strlen(*store)>0) return false; // value is not a number
275
276   return true;
277 };
```

3.4.2.14 bool ftpcGGridSender::readVal (long & value)

Definition at line 258 of file ftpcGGridSender.cc.

```

258
259
260   if(!readAny())return false;
261
262   char* store[256];
263   value=strtol(ptr2,store,10);
264   if(strlen(*store)>0) return false; // value is not a number
265
266   return true;
267 };
```

3.4.2.15 bool ftpcGGridSender::readVal (int & value)

Definition at line 247 of file ftpcGGridSender.cc.

```

247
248
249   if(!readAny()) return false;
250
251   char* store[256];
252   value=(int)strtol(ptr2,store,10);
253   if(strlen(*store)>0) return false; // value is not a number
254
255   return true;
256 };
```

3.4.2.16 bool ftpcGGridSender::readVal (short & value)

Definition at line 236 of file ftpcGGridSender.cc.

```

236
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };

```

3.4.2.17 bool ftpcGGridSender::readVal (double & *value*)

Definition at line 225 of file ftpcGGridSender.cc.

```

225
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };

```

3.4.2.18 bool ftpcGGridSender::readVal (float & *value*)

Definition at line 214 of file ftpcGGridSender.cc.

```

214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };

```

3.4.2.19 bool ftpcGGridSender::readVal (char *& *value*)

Definition at line 207 of file ftpcGGridSender.cc.

```

207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }

```

3.4.2.20 bool ftpcGGridSender::updateDb (const char **fileName*) [virtual]

Definition at line 127 of file ftpcGGridSender.cc.

```

127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     ftpcGGrid* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159 //char mess[256];
160 ostringstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("ftpcGGrid");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175     return true;
176 #undef __METHOD__
177 }
```

3.4.3 Member Data Documentation**3.4.3.1 int ftpcGGridSender::elementList[NUM_DB_ROWS] [protected]**

Definition at line 24 of file ftpcGGridSender.hh.

3.4.3.2 char [ftpcGGridSender::mline\[256\]](#) [protected]

Definition at line 29 of file ftpcGGridSender.hh.

3.4.3.3 bool [ftpcGGridSender::mreadStatus](#) [protected]

Definition at line 28 of file ftpcGGridSender.hh.

3.4.3.4 ftpcGGrid [ftpcGGridSender::previousVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 22 of file ftpcGGridSender.hh.

3.4.3.5 char* [ftpcGGridSender::ptr1](#) [protected]

Definition at line 31 of file ftpcGGridSender.hh.

3.4.3.6 char * [ftpcGGridSender::ptr2](#) [protected]

Definition at line 31 of file ftpcGGridSender.hh.

3.4.3.7 ftcGGrid [ftpcGGridSender::tempVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 23 of file ftpcGGridSender.hh.

3.4.3.8 char [ftpcGGridSender::tmpline\[256\]](#) [protected]

Definition at line 30 of file ftpcGGridSender.hh.

3.4.3.9 int [ftpcGGridSender::updateElements\[NUM_DB_ROWS\]](#) [protected]

Definition at line 26 of file ftpcGGridSender.hh.

3.4.3.10 ftcGGrid [ftpcGGridSender::updateVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 25 of file ftpcGGridSender.hh.

3.4.3.11 float [ftpcGGridSender::vdriftLimit](#) [protected]

dito

Definition at line 36 of file ftpcGGridSender.hh.

The documentation for this class was generated from the following files:

- [ftpcGGridSender.hh](#)
- [ftpcGGridSender.cc](#)
- [ftpcGGridSender_i.cc](#)

3.5 **ftpcHDLCTempsSender** Class Reference

```
#include <ftpcHDLCTempsSender.hh>
```

Public Member Functions

- **ftpcHDLCTempsSender** (const char *localDir)
- virtual ~**ftpcHDLCTempsSender** ()
- virtual void **initTable** ()
- virtual void **initTags** ()
- virtual void **initDataBase** ()
- virtual bool **loadUserControls** (const char *name, const char *value)
- virtual void **initQuery** ()
- virtual bool **queryData** ()
- virtual bool **readData** (const char *fileName)
- virtual bool **updateDb** (const char *fileName)
- virtual bool **readData** (ifstream &from)
- virtual bool **hasChanged** (int rowNumber)
- char * **readAny** ()
- bool **readVal** (char *&value)
- bool **readVal** (float &value)
- bool **readVal** (double &value)
- bool **readVal** (short &value)
- bool **readVal** (int &value)
- bool **readVal** (long &value)
- bool **readVal** (long long &value)
- bool **nextLine** (ifstream &from)
- void **readError** (int l, char *c, char *m)

Protected Attributes

- **ftpcHDLCTemps** **previousVals** [NUM_DB_ROWS]
- **ftpcHDLCTemps** **tempVals** [NUM_DB_ROWS]
- int **elementList** [NUM_DB_ROWS]
- **ftpcHDLCTemps** **updateVals** [NUM_DB_ROWS]
- int **updateElements** [NUM_DB_ROWS]
- bool **mreadStatus**
- char **mline** [256]
- char **tmpline** [256]
- char * **ptr1**
- char * **ptr2**
- float **tdriftLimit**

dito

- float **pdriftLimit**

3.5.1 Constructor & Destructor Documentation

3.5.1.1 **ftpcHDLCTempsSender::ftpcHDLCTempsSender (const char * localDir)**

Definition at line 19 of file ftpcHDLCTempsSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("ftpcHDLCTemps"); // setup the file I/O
24     initDataBase();        // database connections
25     initTable();           // table definitions
26
27 }
```

3.5.1.2 **virtual ftpcHDLCTempsSender::~ftpcHDLCTempsSender () [inline, virtual]**

Definition at line 43 of file ftpcHDLCTempsSender.hh.

```
43 {};
```

3.5.2 Member Function Documentation

3.5.2.1 **bool ftpcHDLCTempsSender::hasChanged (int rowNumber) [virtual]**

Definition at line 126 of file ftpcHDLCTempsSender_i.cc.

```

126
127
128 ftpcHDLCTemps* pre=&previousVals[rowNumber];
129 ftpcHDLCTemps* cur=&tempVals[rowNumber];
130
131     if(fabs(pre->gasOut-cur->gasOut)>=tdriftLimit) return true;
132     if(fabs(pre->waterPressure-cur->waterPressure)>=pdriftLimit) return true;
133     if(fabs(pre->waterIn-cur->waterIn)>=tdriftLimit) return true;
134     if(fabs(pre->waterOut-cur->waterOut)>=tdriftLimit) return true;
135     if(fabs(pre->body1-cur->body1)>=tdriftLimit) return true;
136     if(fabs(pre->body2-cur->body2)>=tdriftLimit) return true;
137     if(fabs(pre->body3-cur->body3)>=tdriftLimit) return true;
138     if(fabs(pre->body4-cur->body4)>=tdriftLimit) return true;
139     if(fabs(pre->body5-cur->body5)>=tdriftLimit) return true;
140     if(fabs(pre->body6-cur->body6)>=tdriftLimit) return true;
141     if(fabs(pre->coolingPlate1-cur->coolingPlate1)>=tdriftLimit) return true;
142     if(fabs(pre->coolingPlate2-cur->coolingPlate2)>=tdriftLimit) return true;
143     if(fabs(pre->coolingPlate3-cur->coolingPlate3)>=tdriftLimit) return true;
144     if(fabs(pre->coolingPlate4-cur->coolingPlate4)>=tdriftLimit) return true;
145     if(fabs(pre->coolingPlate5-cur->coolingPlate5)>=tdriftLimit) return true;
146     if(fabs(pre->coolingPlate6-cur->coolingPlate6)>=tdriftLimit) return true;
147
148 /* example ... note -> change to any element requires db-update
149 * and thus returns true immediately
150 *
151 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
152 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
153 *
154 */
155 */
```

```
157     return false;
158 }
```

3.5.2.2 void ftpcHDLCTempsSender::initDataBase () [virtual]

Definition at line 75 of file ftpcHDLCTempsSender.cc.

```
75 {
76 #define __METHOD__ "initDataBase()"
77
78     /* More than an example... swap user & dbTrg as per subsystem*/
79     mgr->setUser("stardb","");
80     StDbType    dbT = dbConditions;
81     StDbDomain dbD = dbFtpc;
82
83     if( !( node = mgr->initConfig(dbT,dbD) ) )
84         sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.5.2.3 void ftpcHDLCTempsSender::initQuery () [virtual]

Definition at line 44 of file ftpcHDLCTempsSender_i.cc.

```
44 {
45 #define __METHOD__ "initQuery()"
46
47     ofstream to(queryFile);
48
49     if(!to.is_open()){
50         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
51         return;
52     }
53
54     int i;
55     int j;
56     for(i=1; i<3;i++){
57         to<<"ftpc_hdlc:t_gas_"<<i<<.D"<<endl;
58         to<<"ftpc_hdlc:pressure_"<<i<<.D"<<endl;
59         to<<"ftpc_hdlc:t_in_"<<i<<.D"<<endl;
60         to<<"ftpc_hdlc:t_out_"<<i<<.D"<<endl;
61         for(j=1;j<7;j++)to<<"ftpc_hdlc:t_b_"<<i<<_<<j<<.D"<<endl;
62         for(j=1;j<7;j++)to<<"ftpc_hdlc:t_cp_"<<i<<_<<j<<.D"<<endl;
63     }
64 /* example
65 *      for(int i=0;i<16;i++){
66 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.E"<<endl;
67 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.F"<<endl;
68 *          ....
69 *
70 *      }
71 */
72
73     to.close();
74
75 #undef __METHOD__
76 }
```

3.5.2.4 void ftpcHDLCTempsSender::initTable () [virtual]

Definition at line 30 of file ftpcHDLCTempsSender.cc.

```

30
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("ftpcHDLCTemps")))
35         sendMess("Could not find table=ftpcHDLCTemps",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcHDLCTemps));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(ftpcHDLCTemps));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         ftpcHDLCTemps* thv = (ftpcHDLCTemps*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(ftpcHDLCTemps));
55     }
56
57 #undef __METHOD__
58 };

```

3.5.2.5 void ftpcHDLCTempsSender::initTags () [virtual]

Definition at line 66 of file ftpcHDLCTempsSender.cc.

```

66
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("ftpc");
70
71 }

```

3.5.2.6 bool ftpcHDLCTempsSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file ftpcHDLCTempsSender_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 */
26     if(strstr(name,"tdriftLimit")){
27         tdriftLimit=atof(value);
28         sendMess("tdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);

```

```

29     return true;
30 }
31 if(strstr(name,"pdriftLimit")){
32     pdriftLimit=atof(value);
33     sendMess("pdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;
35 }
36
37 return false;
38 #undef __METHOD__
39 }
```

3.5.2.7 bool ftpcHDLCTempsSender::nextLine (ifstream &from) [inline]

Definition at line 76 of file ftpcHDLCTempsSender.hh.

```

76 {
77     if(!from.getline(mline,255))return false;
78     return true;
79 }
```

3.5.2.8 bool ftpcHDLCTempsSender::queryData () [virtual]

Definition at line 91 of file ftpcHDLCTempsSender.cc.

```

91 {
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

3.5.2.9 char * ftpcHDLCTempsSender::readAny ()

Definition at line 197 of file ftpcHDLCTempsSender.cc.

```

197 {
198
199 strcpy(tmpLine,mline);
200 ptr1=tmpLine;
201 ptr2=strtok(ptr1," ");
```

```

202     if(!ptr2) return ptr2;
203     ptr2=strtok(NULL, " ");
204     return ptr2;
205 }
```

3.5.2.10 bool ftpcHDLCTempsSender::readData (ifstream &*from*) [virtual]

Definition at line 81 of file ftpcHDLCTempsSender_i.cc.

```

81 {
82 #define __METHOD__ "readData(ifstream)"
83
84 mreadStatus=true;
85 memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcHDLCTemps));
86
87
88 int i=0;
89 char* c=__CLASS__;
90 char* m=__METHOD__;
91
92     for(int i=0;i<NUM_DB_ROWS;i++){
93         if(!nextLine(from) || !readVal(tempVals[i].gasOut)) readError(__LINE__,c,m);
94         if(!nextLine(from) || !readVal(tempVals[i].waterPressure)) readError(__LINE__,c,m);
95         if(!nextLine(from) || !readVal(tempVals[i].waterIn)) readError(__LINE__,c,m);
96         if(!nextLine(from) || !readVal(tempVals[i].waterOut)) readError(__LINE__,c,m);
97         if(!nextLine(from) || !readVal(tempVals[i].body1)) readError(__LINE__,c,m);
98         if(!nextLine(from) || !readVal(tempVals[i].body2)) readError(__LINE__,c,m);
99         if(!nextLine(from) || !readVal(tempVals[i].body3)) readError(__LINE__,c,m);
100        if(!nextLine(from) || !readVal(tempVals[i].body4)) readError(__LINE__,c,m);
101        if(!nextLine(from) || !readVal(tempVals[i].body5)) readError(__LINE__,c,m);
102        if(!nextLine(from) || !readVal(tempVals[i].body6)) readError(__LINE__,c,m);
103        if(!nextLine(from) || !readVal(tempVals[i].coolingPlate1)) readError(__LINE__,c,m);
104        if(!nextLine(from) || !readVal(tempVals[i].coolingPlate2)) readError(__LINE__,c,m);
105        if(!nextLine(from) || !readVal(tempVals[i].coolingPlate3)) readError(__LINE__,c,m);
106        if(!nextLine(from) || !readVal(tempVals[i].coolingPlate4)) readError(__LINE__,c,m);
107        if(!nextLine(from) || !readVal(tempVals[i].coolingPlate5)) readError(__LINE__,c,m);
108        if(!nextLine(from) || !readVal(tempVals[i].coolingPlate6)) readError(__LINE__,c,m);
109    }
110 /* example format
111 *  for(int i=0;i<NUM_DB_ROWS;i++){
112 *  if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(1,c,m);
113 *
114 *  ....
115 *
116 *  }
117 */
118
119     from.close();
120 return true;
121 #undef __METHOD__
122 }
```

3.5.2.11 bool ftpcHDLCTempsSender::readData (const char **fileName*) [virtual]

Definition at line 116 of file ftpcHDLCTempsSender.cc.

```

116 {
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
```

```

121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.5.2.12 void ftpcHDLCTempsSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 81 of file ftpcHDLCTempsSender.hh.

```

81
82     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
83 }
```

3.5.2.13 bool ftpcHDLCTempsSender::readVal (long long & *value*)

Definition at line 269 of file ftpcHDLCTempsSender.cc.

```

269
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

3.5.2.14 bool ftpcHDLCTempsSender::readVal (long & *value*)

Definition at line 258 of file ftpcHDLCTempsSender.cc.

```

258
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

3.5.2.15 bool ftpcHDLCTempsSender::readVal (int & *value*)

Definition at line 247 of file ftpcHDLCTempsSender.cc.

```

247
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

3.5.2.16 bool ftpcHDLCTempsSender::readVal (short & *value*)

Definition at line 236 of file ftpcHDLCTempsSender.cc.

```
236
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

3.5.2.17 bool ftpcHDLCTempsSender::readVal (double & *value*)

Definition at line 225 of file ftpcHDLCTempsSender.cc.

```
225
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

3.5.2.18 bool ftpcHDLCTempsSender::readVal (float & *value*)

Definition at line 214 of file ftpcHDLCTempsSender.cc.

```
214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

3.5.2.19 bool ftpcHDLCTempsSender::readVal (char *& *value*)

Definition at line 207 of file ftpcHDLCTempsSender.cc.

```
207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

3.5.2.20 bool ftpcHDLCTempsSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file ftpcHDLCTempsSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     ftpcHDLCTemps* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159 //char mess[256];
160 ostringstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("ftpcHDLCTemps");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175     return true;
176 #undef __METHOD__
177 }
```

3.5.3 Member Data Documentation**3.5.3.1 int ftpcHDLCTempsSender::elementList[NUM_DB_ROWS] [protected]**

Definition at line 24 of file ftpcHDLCTempsSender.hh.

3.5.3.2 char [ftpcHDLCTempsSender::mline\[256\]](#) [protected]

Definition at line 29 of file ftpcHDLCTempsSender.hh.

3.5.3.3 bool [ftpcHDLCTempsSender::mreadStatus](#) [protected]

Definition at line 28 of file ftpcHDLCTempsSender.hh.

3.5.3.4 float [ftpcHDLCTempsSender::pdriftLimit](#) [protected]

Definition at line 37 of file ftpcHDLCTempsSender.hh.

3.5.3.5 ftpcHDLCTemps [ftpcHDLCTempsSender::previousVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 22 of file ftpcHDLCTempsSender.hh.

3.5.3.6 char* [ftpcHDLCTempsSender::ptr1](#) [protected]

Definition at line 31 of file ftpcHDLCTempsSender.hh.

3.5.3.7 char * [ftpcHDLCTempsSender::ptr2](#) [protected]

Definition at line 31 of file ftpcHDLCTempsSender.hh.

3.5.3.8 float [ftpcHDLCTempsSender::tdriftLimit](#) [protected]

dito

Definition at line 36 of file ftpcHDLCTempsSender.hh.

3.5.3.9 ftpcHDLCTemps [ftpcHDLCTempsSender::tempVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 23 of file ftpcHDLCTempsSender.hh.

3.5.3.10 char [ftpcHDLCTempsSender::tmpline\[256\]](#) [protected]

Definition at line 30 of file ftpcHDLCTempsSender.hh.

3.5.3.11 int [ftpcHDLCTempsSender::updateElements\[NUM_DB_ROWS\]](#) [protected]

Definition at line 26 of file ftpcHDLCTempsSender.hh.

3.5.3.12 `ftpcHDLCTemps` `ftpcHDLCTempsSender::updateVals[NUM_DB_ROWS]`
[protected]

Definition at line 25 of file `ftpcHDLCTempsSender.hh`.

The documentation for this class was generated from the following files:

- `ftpcHDLCTempsSender.hh`
- `ftpcHDLCTempsSender.cc`
- `ftpcHDLCTempsSender_i.cc`

3.6 fpcHDLCVoltagesSender Class Reference

```
#include <fpcHDLCVoltagesSender.hh>
```

Public Member Functions

- `fpcHDLCVoltagesSender` (const char *localDir)
- virtual `~fpcHDLCVoltagesSender` ()
- virtual void `initTable` ()
- virtual void `initTags` ()
- virtual void `initDataBase` ()
- virtual bool `loadUserControls` (const char *name, const char *value)
- virtual void `initQuery` ()
- virtual bool `queryData` ()
- virtual bool `readData` (const char *fileName)
- virtual bool `updateDb` (const char *fileName)
- virtual bool `readData` (ifstream &from)
- virtual bool `hasChanged` (int rowNumber)
- char * `readAny` ()
- bool `readVal` (char *&value)
- bool `readVal` (float &value)
- bool `readVal` (double &value)
- bool `readVal` (short &value)
- bool `readVal` (int &value)
- bool `readVal` (long &value)
- bool `readVal` (long long &value)
- bool `nextLine` (ifstream &from)
- void `readError` (int l, char *c, char *m)

Protected Attributes

- fpcHDLCVoltages `previousVals` [NUM_DB_ROWS]
- fpcHDLCVoltages `tempVals` [NUM_DB_ROWS]
- int `elementList` [NUM_DB_ROWS]
- fpcHDLCVoltages `updateVals` [NUM_DB_ROWS]
- int `updateElements` [NUM_DB_ROWS]
- bool `mreadStatus`
- char `mline` [256]
- char `tmpline` [256]
- char * `ptr1`
- char * `ptr2`
- float `vdriftLimit`

dito

3.6.1 Constructor & Destructor Documentation

3.6.1.1 ftpcHDLCVoltagesSender::ftpcHDLCVoltagesSender (const char * *localDir*)

Definition at line 19 of file ftpcHDLCVoltagesSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("ftpcHDLCVoltages"); // setup the file I/O
24     initDataBase();          // database connections
25     initTable();             // table definitions
26
27 }
```

3.6.1.2 virtual ftpcHDLCVoltagesSender::~ftpcHDLCVoltagesSender () [inline, virtual]

Definition at line 41 of file ftpcHDLCVoltagesSender.hh.

```
41 {};
```

3.6.2 Member Function Documentation

3.6.2.1 bool ftpcHDLCVoltagesSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 131 of file ftpcHDLCVoltagesSender_i.cc.

```

131
132
133 ftpcHDLCVoltages* pre=&previousVals[rowNumber];
134 ftpcHDLCVoltages* cur=&tempVals[rowNumber];
135
136     if(fabs(pre->v1-cur->v1)>=vdriftLimit) return true;
137     if(fabs(pre->v2-cur->v2)>=vdriftLimit) return true;
138     if(fabs(pre->vcc-cur->vcc)>=vdriftLimit) return true;
139     if(fabs(pre->vee-cur->vee)>=vdriftLimit) return true;
140     if(fabs(pre->vbb-cur->vbb)>=vdriftLimit) return true;
141 /***** removed by MPD 1/27 as per Janet Seyboth*****
142     if(fabs(pre->vdd1-cur->vdd1)>=vdriftLimit) return true;
143     if(fabs(pre->veel-cur->veel)>=vdriftLimit) return true;
144     if(fabs(pre->vdd2-cur->vdd2)>=vdriftLimit) return true;
145     if(fabs(pre->vee2-cur->vee2)>=vdriftLimit) return true;
146     if(fabs(pre->vdd3-cur->vdd3)>=vdriftLimit) return true;
147     if(fabs(pre->vee3-cur->vee3)>=vdriftLimit) return true;
148     if(fabs(pre->vdd4-cur->vdd4)>=vdriftLimit) return true;
149     if(fabs(pre->vee4-cur->vee4)>=vdriftLimit) return true;
150     if(fabs(pre->vdd5-cur->vdd5)>=vdriftLimit) return true;
151     if(fabs(pre->vee5-cur->vee5)>=vdriftLimit) return true;
152 *****/
153
154 /* example ... note -> change to any element requires db-update
155 * and thus returns true immediately
156 *
157 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
158 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
159 *
160 *    ....
```

```

161 */
162
163 return false;
164 }
```

3.6.2.2 void ftpcHDLCVoltagesSender::initDataBase () [virtual]

Definition at line 75 of file ftpcHDLCVoltagesSender.cc.

```

75 {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbFtpc;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.6.2.3 void ftpcHDLCVoltagesSender::initQuery () [virtual]

Definition at line 39 of file ftpcHDLCVoltagesSender_i.cc.

```

39 {
40 #define __METHOD__ "initQuery()"
41
42     ofstream to(queryFile);
43
44     if(!to.is_open()){
45         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
46         return;
47     }
48
49     int i;
50     int j;
51     int k;
52     for(i=1; i<3; i++){
53         for(j=1; j<11; j++) {
54             to<<"ftpc_hdlc:v1_"<<i<<"_ "<<j<<endl;
55             to<<"ftpc_hdlc:v2_"<<i<<"_ "<<j<<endl;
56             to<<"ftpc_hdlc:rdo_vcc_"<<i<<"_ "<<j<<endl;
57             to<<"ftpc_hdlc:rdo_vee_"<<i<<"_ "<<j<<endl;
58             to<<"ftpc_hdlc:rdo_vbb_"<<i<<"_ "<<j<<endl;
59 ****
60         for(k=1;k<6;k++){
61             to<<"ftpc_hdlc:rdo_vdd"<<k<<"_ "<<i<<"_ "<<j<<endl;
62             to<<"ftpc_hdlc:rdo_vee"<<k<<"_ "<<i<<"_ "<<j<<endl;
63         }
64 ****
65
66     }
67 }
68 /* example
69 *      for(int i=0;i<16;i++){
70 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<"_E"<<endl;
71 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<"_F"<<endl;
72 *          ....
```

```

73  *
74  *      }
75  */
76
77     to.close();
78
79 #undef __METHOD__
80 }
```

3.6.2.4 void ftpcHDLCVoltagesSender::initTable () [virtual]

Definition at line 30 of file ftpcHDLCVoltagesSender.cc.

```

30
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("ftpcHDLCVoltages")))
35         sendMess("Could not find table=ftpcHDLCVoltages",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcHDLCVoltages));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(ftpcHDLCVoltages));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         ftpcHDLCVoltages* thv = (ftpcHDLCVoltages*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(ftpcHDLCVoltages));
55     }
56
57 #undef __METHOD__
58 };
```

3.6.2.5 void ftpcHDLCVoltagesSender::initTags () [virtual]

Definition at line 66 of file ftpcHDLCVoltagesSender.cc.

```

66
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("ftpc");
70
71 }
```

3.6.2.6 bool ftpcHDLCVoltagesSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file ftpcHDLCVoltagesSender_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 */
26 if(strstr(name,"vdriftLimit")){
27     vdriftLimit=atof(value);
28     sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
29     return true;
30 }
31
32
33 return false;
34 #undef __METHOD__
35 }
```

3.6.2.7 bool **ftpcHDLCVoltagesSender::nextLine (ifstream & from)** [inline]

Definition at line 74 of file ftpcHDLCVoltagesSender.hh.

```

74
75     if(!from.getline(mline,255))return false;
76     return true;
77 }
```

3.6.2.8 bool **ftpcHDLCVoltagesSender::queryData ()** [virtual]

Definition at line 91 of file ftpcHDLCVoltagesSender.cc.

```

91
92 #define __METHOD__ "queryData( )"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

3.6.2.9 char * **ftpcHDLCVoltagesSender::readAny ()**

Definition at line 197 of file ftpcHDLCVoltagesSender.cc.

```

197
198
199     strcpy(tmpLine,mLine);
200     ptr1=tmpLine;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;
203     ptr2=strtok(NULL," ");
204     return ptr2;
205 }
```

3.6.2.10 bool ftpcHDLCVoltagesSender::readData (ifstream & from) [virtual]

Definition at line 85 of file ftpcHDLCVoltagesSender_i.cc.

```

85
86 #define __METHOD__ "readData(ifstream)"
87
88 mReadStatus=true;
89 memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcHDLCVoltages));
90
91
92 int i=0;
93 char* c=__CLASS__;
94 char* m=__METHOD__;
95
96 for(i=0; i<NUM_DB_ROWS;i++){
97     if(!nextLine(from) || !readVal(tempVals[i].v1)) readError(__LINE__,c,m);
98     if(!nextLine(from) || !readVal(tempVals[i].v2)) readError(__LINE__,c,m);
99     if(!nextLine(from) || !readVal(tempVals[i].vcc)) readError(__LINE__,c,m);
100    if(!nextLine(from) || !readVal(tempVals[i].vee)) readError(__LINE__,c,m);
101    if(!nextLine(from) || !readVal(tempVals[i].vbb)) readError(__LINE__,c,m);
102 /**** removed by MPD 1/27 as per Janet Seyboth*****
103    if(!nextLine(from) || !readVal(tempVals[i].vdd1)) readError(__LINE__,c,m);
104    if(!nextLine(from) || !readVal(tempVals[i].vee1)) readError(__LINE__,c,m);
105    if(!nextLine(from) || !readVal(tempVals[i].vdd2)) readError(__LINE__,c,m);
106    if(!nextLine(from) || !readVal(tempVals[i].vee2)) readError(__LINE__,c,m);
107    if(!nextLine(from) || !readVal(tempVals[i].vdd3)) readError(__LINE__,c,m);
108    if(!nextLine(from) || !readVal(tempVals[i].vee3)) readError(__LINE__,c,m);
109    if(!nextLine(from) || !readVal(tempVals[i].vdd4)) readError(__LINE__,c,m);
110    if(!nextLine(from) || !readVal(tempVals[i].vee4)) readError(__LINE__,c,m);
111    if(!nextLine(from) || !readVal(tempVals[i].vdd5)) readError(__LINE__,c,m);
112    if(!nextLine(from) || !readVal(tempVals[i].vee5)) readError(__LINE__,c,m);
113 *****/
114 }
115 /* example format
116 *  for(int i=0;i<NUM_DB_ROWS;i++){
117 *  if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
118 *
119 *  ....
120 *
121 *  }
122 */
123
124     from.close();
125 return true;
126 #undef __METHOD__
127 }
```

3.6.2.11 bool ftpcHDLCVoltagesSender::readData (const char *fileName) [virtual]

Definition at line 116 of file ftpcHDLCVoltagesSender.cc.

```

116                                     {
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from);    // user implemented file read
123 #undef __METHOD__
124 }
```

3.6.2.12 void ftpcHDLCVoltagesSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 79 of file ftpcHDLCVoltagesSender.hh.

```

79
80     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }
```

3.6.2.13 bool ftpcHDLCVoltagesSender::readVal (long long & *value*)

Definition at line 269 of file ftpcHDLCVoltagesSender.cc.

```

269                                     {
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

3.6.2.14 bool ftpcHDLCVoltagesSender::readVal (long & *value*)

Definition at line 258 of file ftpcHDLCVoltagesSender.cc.

```

258                                     {
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

3.6.2.15 bool ftpcHDLCVoltagesSender::readVal (int & *value*)

Definition at line 247 of file ftpcHDLCVoltagesSender.cc.

```

247                                     {
248
249     if(!readAny()) return false;
```

```

250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };

```

3.6.2.16 bool ftpcHDLCVoltagesSender::readVal (short & *value*)

Definition at line 236 of file ftpcHDLCVoltagesSender.cc.

```

236
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };

```

3.6.2.17 bool ftpcHDLCVoltagesSender::readVal (double & *value*)

Definition at line 225 of file ftpcHDLCVoltagesSender.cc.

```

225
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value= strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };

```

3.6.2.18 bool ftpcHDLCVoltagesSender::readVal (float & *value*)

Definition at line 214 of file ftpcHDLCVoltagesSender.cc.

```

214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };

```

3.6.2.19 bool ftpcHDLCVoltagesSender::readVal (char *& value)

Definition at line 207 of file ftpcHDLCVoltagesSender.cc.

```
207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

3.6.2.20 bool ftpcHDLCVoltagesSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file ftpcHDLCVoltagesSender.cc.

```
127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     ftpcHDLCVoltages* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159 //char mess[256];
160 ostringstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("ftpcHDLCVoltages");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168 if(!mgr->storeDbTable(dbTable)) {
169     addBackLog(writeTime);
170     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171 }
172
173 if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176 #undef __METHOD__
177 }
```

3.6.3 Member Data Documentation

3.6.3.1 int **ftpcHDLCVoltagesSender::elementList[NUM_DB_ROWS]** [protected]

Definition at line 24 of file ftpcHDLCVoltagesSender.hh.

3.6.3.2 char **ftpcHDLCVoltagesSender::mline[256]** [protected]

Definition at line 29 of file ftpcHDLCVoltagesSender.hh.

3.6.3.3 bool **ftpcHDLCVoltagesSender::mreadStatus** [protected]

Definition at line 28 of file ftpcHDLCVoltagesSender.hh.

3.6.3.4 **ftpcHDLCVoltages** **ftpcHDLCVoltagesSender::previousVals[NUM_DB_ROWS]** [protected]

Definition at line 22 of file ftpcHDLCVoltagesSender.hh.

3.6.3.5 char* **ftpcHDLCVoltagesSender::ptr1** [protected]

Definition at line 31 of file ftpcHDLCVoltagesSender.hh.

3.6.3.6 char * **ftpcHDLCVoltagesSender::ptr2** [protected]

Definition at line 31 of file ftpcHDLCVoltagesSender.hh.

3.6.3.7 **ftpcHDLCVoltages** **ftpcHDLCVoltagesSender::tempVals[NUM_DB_ROWS]** [protected]

Definition at line 23 of file ftpcHDLCVoltagesSender.hh.

3.6.3.8 char **ftpcHDLCVoltagesSender::tmpline[256]** [protected]

Definition at line 30 of file ftpcHDLCVoltagesSender.hh.

3.6.3.9 int **ftpcHDLCVoltagesSender::updateElements[NUM_DB_ROWS]** [protected]

Definition at line 26 of file ftpcHDLCVoltagesSender.hh.

3.6.3.10 float [ftpcHDLCVoltages](#) [ftpcHDLCVoltagesSender::updateVals](#)[NUM_DB_ROWS]
[protected]

Definition at line 25 of file ftpcHDLCVoltagesSender.hh.

3.6.3.11 float [ftpcHDLCVoltages](#) [ftpcHDLCVoltagesSender::vdriftLimit](#) [protected]

dito

Definition at line 35 of file ftpcHDLCVoltagesSender.hh.

The documentation for this class was generated from the following files:

- [ftpcHDLCVoltagesSender.hh](#)
- [ftpcHDLCVoltagesSender.cc](#)
- [ftpcHDLCVoltagesSender_i.cc](#)

3.7 **ftpcTempsSender** Class Reference

```
#include <ftpcTempsSender.hh>
```

Public Member Functions

- **ftpcTempsSender** (const char *localDir)
- virtual ~**ftpcTempsSender** ()
- virtual void **initTable** ()
- virtual void **initTags** ()
- virtual void **initDataBase** ()
- virtual bool **loadUserControls** (const char *name, const char *value)
- virtual void **initQuery** ()
- virtual bool **queryData** ()
- virtual bool **readData** (const char *fileName)
- virtual bool **updateDb** (const char *fileName)
- virtual bool **readData** (ifstream &from)
- virtual bool **hasChanged** (int rowNumber)
- char * **readAny** ()
- bool **readVal** (char *&value)
- bool **readVal** (float &value)
- bool **readVal** (double &value)
- bool **readVal** (short &value)
- bool **readVal** (int &value)
- bool **readVal** (long &value)
- bool **readVal** (long long &value)
- bool **nextLine** (ifstream &from)
- void **readError** (int l, char *c, char *m)

Protected Attributes

- **ftpcTemps previousVals** [NUM_DB_ROWS]
- **ftpcTemps tempVals** [NUM_DB_ROWS]
- int **elementList** [NUM_DB_ROWS]
- **ftpcTemps updateVals** [NUM_DB_ROWS]
- int **updateElements** [NUM_DB_ROWS]
- bool **mreadStatus**
- char **mline** [256]
- char **tmpline** [256]
- char * **ptr1**
- char * **ptr2**
- float **extdriftLimit**

dito

3.7.1 Constructor & Destructor Documentation

3.7.1.1 **ftpcTempsSender::ftpcTempsSender (const char * localDir)**

Definition at line 19 of file ftpcTempsSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("ftpcTemps"); // setup the file I/O
24     initDataBase();      // database connections
25     initTable();         // table definitions
26
27 }
```

3.7.1.2 **virtual ftpcTempsSender::~ftpcTempsSender () [inline, virtual]**

Definition at line 41 of file ftpcTempsSender.hh.

```
41 {};
```

3.7.2 Member Function Documentation

3.7.2.1 **bool ftpcTempsSender::hasChanged (int rowCount) [virtual]**

Definition at line 119 of file ftpcTempsSender_i.cc.

```

119
120
121 ftpcTemps* pre=&previousVals[rowNumber];
122 ftpcTemps* cur=&tempVals[rowNumber];
123
124 if(fabs(pre->extra1East-cur->extra1East)>=extdriftLimit) return true;
125 if(fabs(pre->extra2East-cur->extra2East)>=extdriftLimit) return true;
126 if(fabs(pre->extra3East-cur->extra3East)>=extdriftLimit) return true;
127 if(fabs(pre->extra4East-cur->extra4East)>=extdriftLimit) return true;
128 if(fabs(pre->extra5East-cur->extra5East)>=extdriftLimit) return true;
129 if(fabs(pre->extra6East-cur->extra6East)>=extdriftLimit) return true;
130 if(fabs(pre->extra7East-cur->extra7East)>=extdriftLimit) return true;
131 if(fabs(pre->extra1West-cur->extra1West)>=extdriftLimit) return true;
132 if(fabs(pre->extra2West-cur->extra2West)>=extdriftLimit) return true;
133 if(fabs(pre->extra3West-cur->extra3West)>=extdriftLimit) return true;
134 if(fabs(pre->extra4West-cur->extra4West)>=extdriftLimit) return true;
135 if(fabs(pre->extra5West-cur->extra5West)>=extdriftLimit) return true;
136 if(fabs(pre->extra6West-cur->extra6West)>=extdriftLimit) return true;
137 if(fabs(pre->extra7West-cur->extra7West)>=extdriftLimit) return true;
138
139 /* example ... note -> change to any element requires db-update
140 * and thus returns true immediately
141 *
142 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
143 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
144 *
145 * ....
146 */
147
148 return false;
149 }
```

3.7.2.2 void ftpcTempsSender::initDataBase () [virtual]

Definition at line 75 of file ftpcTempsSender.cc.

```

75
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbFtpc;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.7.2.3 void ftpcTempsSender::initQuery () [virtual]

Definition at line 38 of file ftpcTempsSender_i.cc.

```

38
39 #define __METHOD__ "initQuery()"
40
41     ofstream to(queryFile);
42
43     if(!to.is_open()){
44         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
45         return;
46     }
47
48     int i;
49     for(i=1;i<8;i++)
50     {
51         to<<"ftpc_t_ex_1_"<<i<<.VAL"\  
endl;
52         to<<"ftpc_t_ex_2_"<<i<<.VAL"\  
endl;
53     }
54
55
56 /* example
57 *      for(int i=0;i<16;i++){
58 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.E"\  
endl;
59 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.F"\  
endl;
60 *          ....
61 *
62 *      }
63 */
64
65     to.close();
66
67 #undef __METHOD__
68 }
```

3.7.2.4 void ftpcTempsSender::initTable () [virtual]

Definition at line 30 of file ftpcTempsSender.cc.

```
30 {
```

```

31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(! (table=node->addDbTable("ftpcTemps")))
35         sendMess("Could not find table=ftpcTemps",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcTemps));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(ftpcTemps));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         ftpcTemps* thv = (ftpcTemps*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(ftpcTemps));
55     }
56
57 #undef __METHOD__
58 };

```

3.7.2.5 void ftpcTempsSender::initTags () [virtual]

Definition at line 66 of file ftpcTempsSender.cc.

```

66 {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("ftpc");
70 }
71 }

```

3.7.2.6 bool ftpcTempsSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file ftpcTempsSender_i.cc.

```

20 {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 // more than an example ... swap driftLimit to yours
24 // and duplicate this structure for each selection criteria
25     if(strstr(name,"extdriftLimit")){
26         extdriftLimit=atof(value);
27         sendMess("extdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28         return true;
29     }
30
31
32 return false;
33 #undef __METHOD__
34 }

```

3.7.2.7 bool ftpcTempsSender::nextLine (ifstream &from) [inline]

Definition at line 74 of file ftpcTempsSender.hh.

```
74
75     if(!from.getline(mline,255))return false;
76     return true;
77 }
```

3.7.2.8 bool ftpcTempsSender::queryData () [virtual]

Definition at line 91 of file ftpcTempsSender.cc.

```
91
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

3.7.2.9 char * ftpcTempsSender::readAny ()

Definition at line 206 of file ftpcTempsSender.cc.

```
206
207
208 strcpy(tmpline,mline);
209 ptr1=tmpline;
210 ptr2=strtok(ptr1," ");
211 if(!ptr2) return ptr2;
212 ptr2=strtok(NULL," ");
213 return ptr2;
214 }
```

3.7.2.10 bool ftpcTempsSender::readData (ifstream &from) [virtual]

Definition at line 73 of file ftpcTempsSender_i.cc.

```
73
74 #define __METHOD__ "readData(ifstream)"
```

```

75
76 mreadStatus=true;
77 memset(tempVals,0,NUM_DB_ROWS*sizeof(ftpcTemps));
78
79
80 int i=0;
81 char* c=__CLASS__;
82 char* m=__METHOD__;
83
84 for(int i=0;i<NUM_DB_ROWS;i++){
85     if(!nextLine(from) || !readVal(tempVals[i].extra1West)) readError(__LINE__,c,m);
86     if(!nextLine(from) || !readVal(tempVals[i].extra1East)) readError(__LINE__,c,m);
87     if(!nextLine(from) || !readVal(tempVals[i].extra2West)) readError(__LINE__,c,m);
88     if(!nextLine(from) || !readVal(tempVals[i].extra2East)) readError(__LINE__,c,m);
89     if(!nextLine(from) || !readVal(tempVals[i].extra3West)) readError(__LINE__,c,m);
90     if(!nextLine(from) || !readVal(tempVals[i].extra3East)) readError(__LINE__,c,m);
91     if(!nextLine(from) || !readVal(tempVals[i].extra4West)) readError(__LINE__,c,m);
92     if(!nextLine(from) || !readVal(tempVals[i].extra4East)) readError(__LINE__,c,m);
93     if(!nextLine(from) || !readVal(tempVals[i].extra5West)) readError(__LINE__,c,m);
94     if(!nextLine(from) || !readVal(tempVals[i].extra5East)) readError(__LINE__,c,m);
95     if(!nextLine(from) || !readVal(tempVals[i].extra6West)) readError(__LINE__,c,m);
96     if(!nextLine(from) || !readVal(tempVals[i].extra6East)) readError(__LINE__,c,m);
97     if(!nextLine(from) || !readVal(tempVals[i].extra7West)) readError(__LINE__,c,m);
98     if(!nextLine(from) || !readVal(tempVals[i].extra7East)) readError(__LINE__,c,m);
99 }
100
101
102 cout<<"HEEEEEEEEEEEEREEEEEEE\n";
103 /* example format
104 *  for(int i=0;i<NUM_DB_ROWS;i++){
105 *    if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
106 *
107 *    ....
108 *
109 *  }
110 */
111
112 from.close();
113 return true;
114 #undef __METHOD__
115 }
```

3.7.2.11 bool ftpcTempsSender::readData (const char *fileName) [virtual]

Definition at line 116 of file ftpcTempsSender.cc.

```

116
117 #define __METHOD__ "readData(fileName)"
118
119   ifstream from(fileName);
120   if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122   return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.7.2.12 void ftpcTempsSender::readError (int l, char * c, char * m) [inline]

Definition at line 79 of file ftpcTempsSender.hh.

```
80     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }
```

3.7.2.13 bool ftpcTempsSender::readVal (long long & value)

Definition at line 278 of file ftpcTempsSender.cc.

```
278
279
280     if(!readAny())return false;
281     char* store[256];
282     value=strtoll(ptr2,store,10);
283     if(strlen(*store)>0) return false; // value is not a number
284
285     return true;
286 };
```

3.7.2.14 bool ftpcTempsSender::readVal (long & value)

Definition at line 267 of file ftpcTempsSender.cc.

```
267
268
269     if(!readAny())return false;
270
271     char* store[256];
272     value=strtol(ptr2,store,10);
273     if(strlen(*store)>0) return false; // value is not a number
274
275     return true;
276 };
```

3.7.2.15 bool ftpcTempsSender::readVal (int & value)

Definition at line 256 of file ftpcTempsSender.cc.

```
256
257
258     if(!readAny()) return false;
259
260     char* store[256];
261     value=(int)strtol(ptr2,store,10);
262     if(strlen(*store)>0) return false; // value is not a number
263
264     return true;
265 };
```

3.7.2.16 bool ftpcTempsSender::readVal (short & value)

Definition at line 245 of file ftpcTempsSender.cc.

```
245
246
247     if(!readAny()) return false;
```

```

248
249     char* store[256];
250     value=(short)strtol(ptr2,store,10);
251     if(strlen(*store)>0) return false; // value is not a number
252
253     return true;
254 };

```

3.7.2.17 bool ftpcTempsSender::readVal (double & *value*)

Definition at line 234 of file ftpcTempsSender.cc.

```

234
235
236     if(!readAny())return false;
237
238     char* store[256];
239     value=strtod(ptr2,store);
240     if(strlen(*store)>0) return false; // value is not a number
241
242     return true;
243 };

```

3.7.2.18 bool ftpcTempsSender::readVal (float & *value*)

Definition at line 223 of file ftpcTempsSender.cc.

```

223
224
225     if(!readAny()) return false;
226
227     char* store[256];
228     value=(float)strtod(ptr2,store);
229     if(strlen(*store)>0) return false; // value is not a number
230
231     return true;
232 };

```

3.7.2.19 bool ftpcTempsSender::readVal (char *& *value*)

Definition at line 216 of file ftpcTempsSender.cc.

```

216
217
218     if(!readAny()) return false;
219     strcpy(value,ptr2);
220     return true;
221 }

```

3.7.2.20 bool ftpcTempsSender::updateDb (const char **fileName*) [virtual]

Definition at line 127 of file ftpcTempsSender.cc.

```

127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     ftpcTemps* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mibName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159     //char mess[256];
160     ostringstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("ftpcTemps");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168 //MPD debug
169 //ostringstream mpd;
170 //mpd<<"***** "<<dbTable<<" "<<writeTime<<" " <<node<<ends;
171 //sendMess((mpd.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
172
173     if(!mgr->storeDbTable(dbTable)) {
174         addBackLog(writeTime);
175         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
176     }
177
178     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
179 //MPD debug
180 //ostringstream mpd2;
181 //mpd2<<"***** "<<dbTable<<" "<<writeTime<<" " <<node<<ends;
182 //sendMess((mpd2.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
183
184     return true;
185 #undef __METHOD__
186 }
```

3.7.3 Member Data Documentation

3.7.3.1 int `ftpcTempsSender::elementList[NUM_DB_ROWS]` [protected]

Definition at line 24 of file ftpcTempsSender.hh.

3.7.3.2 float `ftpcTempsSender::extdriftLimit` [protected]

dito

Definition at line 35 of file ftpcTempsSender.hh.

3.7.3.3 char `ftpcTempsSender::mline[256]` [protected]

Definition at line 29 of file ftpcTempsSender.hh.

3.7.3.4 bool `ftpcTempsSender::mreadStatus` [protected]

Definition at line 28 of file ftpcTempsSender.hh.

3.7.3.5 `ftpcTemps` `ftpcTempsSender::previousVals[NUM_DB_ROWS]` [protected]

Definition at line 22 of file ftpcTempsSender.hh.

3.7.3.6 char* `ftpcTempsSender::ptr1` [protected]

Definition at line 31 of file ftpcTempsSender.hh.

3.7.3.7 char * `ftpcTempsSender::ptr2` [protected]

Definition at line 31 of file ftpcTempsSender.hh.

3.7.3.8 `ftpcTemps` `ftpcTempsSender::tempVals[NUM_DB_ROWS]` [protected]

Definition at line 23 of file ftpcTempsSender.hh.

3.7.3.9 char `ftpcTempsSender::tmpline[256]` [protected]

Definition at line 30 of file ftpcTempsSender.hh.

3.7.3.10 int `ftpcTempsSender::updateElements[NUM_DB_ROWS]` [protected]

Definition at line 26 of file ftpcTempsSender.hh.

3.7.3.11 ftpcTemps **ftpcTempsSender::updateVals[NUM_DB_ROWS**]** [protected]**

Definition at line 25 of file **ftpcTempsSender.hh**.

The documentation for this class was generated from the following files:

- [ftpcTempsSender.hh](#)
- [ftpcTempsSender.cc](#)
- [ftpcTempsSender_i.cc](#)

Chapter 4

Doxxygen_MPDAEMON File Documentation

4.1 ftpcAnodesDaemon.cc File Reference

```
#include "ftpcAnodesSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char **ldir*)

4.1.1 Function Documentation

4.1.1.1 void [runSender](#) (const char * *ldir*)

Definition at line 14 of file ftpcAnodesDaemon.cc.

```
14 {  
15  
16     CndDbSender* sender = new ftpcAnodesSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24 };
```

4.2 ftpcAnodesSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "ftpcAnodesSender.hh"
#include "StDbTable.h"
#include "ftpcAnodesSender_i.cc"
```

Defines

- #define __CLASS__ "ftpcAnodesSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.2.1 Define Documentation

4.2.1.1 #define __CLASS__ "ftpcAnodesSender"

Definition at line 17 of file ftpcAnodesSender.cc.

4.2.1.2 #define __METHOD__ "updateDb(filename)"

4.2.1.3 #define __METHOD__ "readData(fileName)"

4.2.1.4 #define __METHOD__ "queryData()"

4.2.1.5 #define __METHOD__ "initDataBase()"

4.2.1.6 #define __METHOD__ "initTable()"

4.3 **ftpcAnodesSender.hh** File Reference

```
#include "CndDbSender.hh"
#include "ftpcAnodes.h"
```

Classes

- class [ftpcAnodesSender](#)

Defines

- #define [NUM_DB_ROWS](#) 2

4.3.1 Define Documentation

4.3.1.1 #define NUM_DB_ROWS 2

Definition at line 16 of file ftpcAnodesSender.hh.

4.4 ftpcAnodesSender_i.cc File Reference

Defines

- #define METHOD_ "loadUserControls(name,value)"
- #define METHOD_ "initQuery()"
- #define METHOD_ "readData(ifstream)"

4.4.1 Define Documentation

4.4.1.1 #define METHOD_ "readData(ifstream)"

4.4.1.2 #define METHOD_ "initQuery()"

4.4.1.3 #define METHOD_ "loadUserControls(name,value)"

4.5 ftpcCathodeDaemon.cc File Reference

```
#include "ftpcCathodeSender.hh"
#include <unistd.h>
```

Functions

- void **runSender** (const char *ldir)

4.5.1 Function Documentation

4.5.1.1 void runSender (const char * *ldir*)

Definition at line 14 of file ftpcCathodeDaemon.cc.

```
14
15
16     CndDbSender* sender = new ftpcCathodeSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.6 ftpcCathodeSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "ftpcCathodeSender.hh"
#include "StDbTable.h"
#include "ftpcCathodeSender_i.cc"
```

Defines

- #define __CLASS__ "ftpcCathodeSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.6.1 Define Documentation

4.6.1.1 #define __CLASS__ "ftpcCathodeSender"

Definition at line 17 of file ftpcCathodeSender.cc.

4.6.1.2 #define __METHOD__ "updateDb(filename)"

4.6.1.3 #define __METHOD__ "readData(fileName)"

4.6.1.4 #define __METHOD__ "queryData()"

4.6.1.5 #define __METHOD__ "initDataBase()"

4.6.1.6 #define __METHOD__ "initTable()"

4.7 ftpcCathodeSender.hh File Reference

```
#include "CndDbSender.hh"
#include "ftpcCathode.h"
```

Classes

- class [ftpcCathodeSender](#)

Defines

- #define [NUM_DB_ROWS](#) 2

4.7.1 Define Documentation

4.7.1.1 #define NUM_DB_ROWS 2

Definition at line 16 of file ftpcCathodeSender.hh.

4.8 fpcCathodeSender_i.cc File Reference

Defines

- #define METHOD_ "loadUserControls(name,value)"
- #define METHOD_ "initQuery()"
- #define METHOD_ "readData(ifstream)"

4.8.1 Define Documentation

4.8.1.1 #define METHOD_ "readData(ifstream)"

4.8.1.2 #define METHOD_ "initQuery()"

4.8.1.3 #define METHOD_ "loadUserControls(name,value)"

4.9 ftpcGasSystemDaemon.cc File Reference

```
#include "ftpcGasSystemSender.hh"
#include <unistd.h>
```

Functions

- void **runSender** (const char *ldir)

4.9.1 Function Documentation

4.9.1.1 void runSender (const char * *ldir*)

Definition at line 14 of file ftpcGasSystemDaemon.cc.

```
14
15
16     CndDbSender* sender = new ftpcGasSystemSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.10 ftpcGasSystemSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "ftpcGasSystemSender.hh"
#include "StDbTable.h"
#include "ftpcGasSystemSender_i.cc"
```

Defines

- #define __CLASS__ "ftpcGasSystemSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.10.1 Define Documentation

4.10.1.1 #define __CLASS__ "ftpcGasSystemSender"

Definition at line 17 of file ftpcGasSystemSender.cc.

4.10.1.2 #define __METHOD__ "updateDb(filename)"

4.10.1.3 #define __METHOD__ "readData(fileName)"

4.10.1.4 #define __METHOD__ "queryData()"

4.10.1.5 #define __METHOD__ "initDataBase()"

4.10.1.6 #define __METHOD__ "initTable()"

4.11 ftpcGasSystemSender.hh File Reference

```
#include "CndDbSender.hh"
#include "ftpcGasSystem.h"
```

Classes

- class [ftpcGasSystemSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

4.11.1 Define Documentation

4.11.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file ftpcGasSystemSender.hh.

4.12 ft pcGasSystemSender_i.cc File Reference

Defines

- #define METHOD "loadUserControls(name,value)"
- #define METHOD "initQuery()"
- #define METHOD "readData(ifstream)"

4.12.1 Define Documentation

4.12.1.1 #define METHOD "readData(ifstream)"

4.12.1.2 #define METHOD "initQuery()"

4.12.1.3 #define METHOD "loadUserControls(name,value)"

4.13 ftpcGGridDaemon.cc File Reference

```
#include "ftpcGGridSender.hh"
#include <unistd.h>
```

Functions

- void **runSender** (const char **ldir*)

4.13.1 Function Documentation

4.13.1.1 void runSender (const char * *ldir*)

Definition at line 14 of file ftpcGGridDaemon.cc.

```
14
15
16     CndDbSender* sender = new ftpcGGridSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.14 ftpcGGridSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "ftpcGGridSender.hh"
#include "StDbTable.h"
#include "ftpcGGridSender_i.cc"
```

Defines

- #define __CLASS__ "ftpcGGridSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.14.1 Define Documentation

4.14.1.1 #define __CLASS__ "ftpcGGridSender"

Definition at line 17 of file ftpcGGridSender.cc.

4.14.1.2 #define __METHOD__ "updateDb(filename)"

4.14.1.3 #define __METHOD__ "readData(fileName)"

4.14.1.4 #define __METHOD__ "queryData()"

4.14.1.5 #define __METHOD__ "initDataBase()"

4.14.1.6 #define __METHOD__ "initTable()"

4.15 ft pcGGridSender.hh File Reference

```
#include "CndDbSender.hh"
#include "ft pcGGrid.h"
```

Classes

- class [ft pcGGridSender](#)

Defines

- #define [NUM_DB_ROWS](#) 2

4.15.1 Define Documentation

4.15.1.1 #define NUM_DB_ROWS 2

Definition at line 16 of file ft pcGGridSender.hh.

4.16 ftpcGGridSender_i.cc File Reference

Defines

- #define METHOD "loadUserControls(name,value)"
- #define METHOD "initQuery()"
- #define METHOD "readData(ifstream)"

4.16.1 Define Documentation

4.16.1.1 #define METHOD "readData(ifstream)"

4.16.1.2 #define METHOD "initQuery()"

4.16.1.3 #define METHOD "loadUserControls(name,value)"

4.17 ftpcHDLCTempsDaemon.cc File Reference

```
#include "ftpcHDLCTempsSender.hh"
#include <unistd.h>
```

Functions

- void **runSender** (const char *ldir)

4.17.1 Function Documentation

4.17.1.1 void **runSender** (const char * *ldir*)

Definition at line 14 of file ftpcHDLCTempsDaemon.cc.

```
14
15
16     CndDbSender* sender = new ftpcHDLCTempsSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.18 ftpcHDLCTempsSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "ftpcHDLCTempsSender.hh"
#include "StDbTable.h"
#include "ftpcHDLCTempsSender_i.cc"
```

Defines

- #define __CLASS__ "ftpcHDLCTempsSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.18.1 Define Documentation

4.18.1.1 #define __CLASS__ "ftpcHDLCTempsSender"

Definition at line 17 of file ftpcHDLCTempsSender.cc.

4.18.1.2 #define __METHOD__ "updateDb(filename)"

4.18.1.3 #define __METHOD__ "readData(fileName)"

4.18.1.4 #define __METHOD__ "queryData()"

4.18.1.5 #define __METHOD__ "initDataBase()"

4.18.1.6 #define __METHOD__ "initTable()"

4.19 ftpcHDLCTempsSender.hh File Reference

```
#include "CndDbSender.hh"
#include "ftpcHDLCTemps.h"
```

Classes

- class [ftpcHDLCTempsSender](#)

Defines

- #define [NUM_DB_ROWS](#) 2

4.19.1 Define Documentation

4.19.1.1 #define NUM_DB_ROWS 2

Definition at line 16 of file ftpcHDLCTempsSender.hh.

4.20 ftpcHDLCTempsSender_i.cc File Reference

Defines

- #define METHOD "loadUserControls(name,value)"
- #define METHOD "initQuery()"
- #define METHOD "readData(ifstream)"

4.20.1 Define Documentation

4.20.1.1 #define METHOD "readData(ifstream)"

4.20.1.2 #define METHOD "initQuery()"

4.20.1.3 #define METHOD "loadUserControls(name,value)"

4.21 ftpcHDLCVoltagesDaemon.cc File Reference

```
#include "ftpcHDLCVoltagesSender.hh"
#include <unistd.h>
```

Functions

- void **runSender** (const char **ldir*)

4.21.1 Function Documentation

4.21.1.1 void runSender (const char * *ldir*)

Definition at line 14 of file ftpcHDLCVoltagesDaemon.cc.

```
14
15
16     CndDbSender* sender = new ftpcHDLCVoltagesSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.22 ftpcHDLCVoltagesSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "ftpcHDLCVoltagesSender.hh"
#include "StDbTable.h"
#include "ftpcHDLCVoltagesSender_i.cc"
```

Defines

- #define __CLASS__ "ftpcHDLCVoltagesSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.22.1 Define Documentation

4.22.1.1 #define __CLASS__ "ftpcHDLCVoltagesSender"

Definition at line 17 of file ftpcHDLCVoltagesSender.cc.

4.22.1.2 #define __METHOD__ "updateDb(filename)"

4.22.1.3 #define __METHOD__ "readData(fileName)"

4.22.1.4 #define __METHOD__ "queryData()"

4.22.1.5 #define __METHOD__ "initDataBase()"

4.22.1.6 #define __METHOD__ "initTable()"

4.23 ftpcHDLCVoltagesSender.hh File Reference

```
#include "CndDbSender.hh"
#include "ftpcHDLCVoltages.h"
```

Classes

- class [ftpcHDLCVoltagesSender](#)

Defines

- #define [NUM_DB_ROWS](#) 20

4.23.1 Define Documentation

4.23.1.1 #define NUM_DB_ROWS 20

Definition at line 16 of file ftpcHDLCVoltagesSender.hh.

4.24 ftpcHDLCVoltagesSender_i.cc File Reference

Defines

- #define METHOD_ "loadUserControls(name,value)"
- #define METHOD_ "initQuery()"
- #define METHOD_ "readData(ifstream)"

4.24.1 Define Documentation

4.24.1.1 #define METHOD_ "readData(ifstream)"

4.24.1.2 #define METHOD_ "initQuery()"

4.24.1.3 #define METHOD_ "loadUserControls(name,value)"

4.25 ftpcTempsDaemon.cc File Reference

```
#include "ftpcTempsSender.hh"
#include <unistd.h>
```

Functions

- void **runSender** (const char *ldir)

4.25.1 Function Documentation

4.25.1.1 void runSender (const char * *ldir*)

Definition at line 14 of file ftpcTempsDaemon.cc.

```
14
15
16     CndDbSender* sender = new ftpcTempsSender(ldir);
17
18     sender->initQuery();
19     for(;;) { //ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.26 ftpcTempsSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "ftpcTempsSender.hh"
#include "StDbTable.h"
#include "ftpcTempsSender_i.cc"
```

Defines

- #define __CLASS__ "ftpcTempsSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.26.1 Define Documentation

4.26.1.1 #define __CLASS__ "ftpcTempsSender"

Definition at line 17 of file ftpcTempsSender.cc.

4.26.1.2 #define __METHOD__ "updateDb(filename)"

4.26.1.3 #define __METHOD__ "readData(fileName)"

4.26.1.4 #define __METHOD__ "queryData()"

4.26.1.5 #define __METHOD__ "initDataBase()"

4.26.1.6 #define __METHOD__ "initTable()"

4.27 ftpcTempsSender.hh File Reference

```
#include "CndDbSender.hh"
#include "ftpcTemps.h"
```

Classes

- class [ftpcTempsSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

4.27.1 Define Documentation

4.27.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file ftpcTempsSender.hh.

4.28 ftpcTempsSender_i.cc File Reference

Defines

- #define METHOD "loadUserControls(name,value)"
- #define METHOD "initQuery()"
- #define METHOD "readData(ifstream)"

4.28.1 Define Documentation

4.28.1.1 #define METHOD "readData(ifstream)"

4.28.1.2 #define METHOD "initQuery()"

4.28.1.3 #define METHOD "loadUserControls(name,value)"

Index

~ftpcAnodesSender
 ftpcAnodesSender, 6

~ftpcCathodeSender
 ftpcCathodeSender, 17

~ftpcGGridSender
 ftpcGGridSender, 38

~ftpcGasSystemSender
 ftpcGasSystemSender, 27

~ftpcHDLCTempsSender
 ftpcHDLCTempsSender, 48

~ftpcHDLCVoltagesSender
 ftpcHDLCVoltagesSender, 59

~ftpcTempsSender
 ftpcTempsSender, 70

__CLASS__
 ftpcAnodesSender.cc, 82
 ftpcCathodeSender.cc, 86
 ftpcGasSystemSender.cc, 90
 ftpcGGridSender.cc, 94
 ftpcHDLCTempsSender.cc, 98
 ftpcHDLCVoltagesSender.cc, 102
 ftpcTempsSender.cc, 106

__METHOD__
 ftpcAnodesSender.cc, 82
 ftpcAnodesSender_i.cc, 84
 ftpcCathodeSender.cc, 86
 ftpcCathodeSender_i.cc, 88
 ftpcGasSystemSender.cc, 90
 ftpcGasSystemSender_i.cc, 92
 ftpcGGridSender.cc, 94
 ftpcGGridSender_i.cc, 96
 ftpcHDLCTempsSender.cc, 98
 ftpcHDLCTempsSender_i.cc, 100
 ftpcHDLCVoltagesSender.cc, 102
 ftpcHDLCVoltagesSender_i.cc, 104
 ftpcTempsSender.cc, 106
 ftpcTempsSender_i.cc, 108

cdriftLimit
 ftpcAnodesSender, 14
 ftpcCathodeSender, 24

elementList
 ftpcAnodesSender, 14
 ftpcCathodeSender, 24

ftpcGasSystemSender, 34

ftpcGGridSender, 45

ftpcHDLCTempsSender, 55

ftpcHDLCVoltagesSender, 67

ftpcTempsSender, 78

extdriftLimit
 ftpcTempsSender, 78

flowdriftLimit
 ftpcGasSystemSender, 34

ftpcAnodesDaemon.cc, 81

ftpcAnodesDaemon.cc
 runSender, 81

ftpcAnodesSender, 5
 ftpcAnodesSender, 6

ftpcAnodesSender
 ~ftpcAnodesSender, 6
 cdriftLimit, 14
 elementList, 14
 ftpcAnodesSender, 6
 hasChanged, 6
 initDataBase, 7
 initQuery, 7
 initTable, 8
 initTags, 8
 loadUserControls, 8
 mline, 14
 mreadStatus, 14
 nextLine, 9
 previousVals, 14
 ptr1, 14
 ptr2, 14
 queryData, 9
 readAny, 10
 readData, 10
 readError, 11
 readVal, 11, 12
 tempVals, 14
 tmpline, 14
 updateDb, 13
 updateElements, 14
 updateVals, 14
 vdriftLimit, 15

ftpcAnodesSender.cc, 82

ftpcAnodesSender.cc

__CLASS__, 82
 __METHOD__, 82
 ftpcAnodesSender.hh, 83
 ftpcAnodesSender.hh
 NUM_DB_ROWS, 83
 ftpcAnodesSender_i.cc, 84
 ftpcAnodesSender_i.cc
 __METHOD__, 84
 ftpcCathodeDaemon.cc, 85
 ftpcCathodeDaemon.cc
 runSender, 85
 ftpcCathodeSender, 16
 ftpcCathodeSender, 17
 ftpcCathodeSender
 ~ftpcCathodeSender, 17
 cdriftLimit, 24
 elementList, 24
 ftpcCathodeSender, 17
 hasChanged, 17
 initDataBase, 17
 initQuery, 18
 initTable, 18
 initTags, 19
 loadUserControls, 19
 mline, 24
 mreadStatus, 24
 nextLine, 19
 previousVals, 24
 ptr1, 24
 ptr2, 24
 queryData, 20
 readAny, 20
 readData, 20, 21
 readError, 21
 readVal, 21–23
 tempVals, 24
 tmpline, 25
 updateDb, 23
 updateElements, 25
 updateVals, 25
 vdriftLimit, 25
 ftpcCathodeSender.cc, 86
 ftpcCathodeSender.cc
 __CLASS__, 86
 __METHOD__, 86
 ftpcCathodeSender.hh, 87
 ftpcCathodeSender.hh
 NUM_DB_ROWS, 87
 ftpcCathodeSender_i.cc, 88
 ftpcCathodeSender_i.cc
 __METHOD__, 88
 ftpcGasSystemDaemon.cc, 89
 ftpcGasSystemDaemon.cc
 runSender, 89
 ftpcGasSystemSender, 26
 ~ftpcGasSystemSender, 27
 ftpcGasSystemSender
 ~ftpcGasSystemSender, 27
 elementList, 34
 flowdriftLimit, 34
 ftpcGasSystemSender, 27
 h2odriftLimit, 35
 hasChanged, 27
 initDataBase, 27
 initQuery, 28
 initTable, 28
 initTags, 29
 loadUserControls, 29
 mline, 35
 mreadStatus, 35
 nextLine, 30
 o2driftLimit, 35
 previousVals, 35
 ptr1, 35
 ptr2, 35
 queryData, 30
 readAny, 30
 readData, 30, 31
 readError, 31
 readVal, 32, 33
 tempVals, 35
 tmpline, 35
 updateDb, 33
 updateElements, 35
 updateVals, 35
 ftpcGasSystemSender.cc, 90
 ftpcGasSystemSender.cc
 __CLASS__, 90
 __METHOD__, 90
 ftpcGasSystemSender.hh, 91
 ftpcGasSystemSender.hh
 NUM_DB_ROWS, 91
 ftpcGasSystemSender_i.cc, 92
 ftpcGasSystemSender_i.cc
 __METHOD__, 92
 ftpcGGridDaemon.cc, 93
 ftpcGGridDaemon.cc
 runSender, 93
 ftpcGGridSender, 37
 ftpcGGridSender, 38
 ftpcGGridSender
 ~ftpcGGridSender, 38
 elementList, 45
 ftpcGGridSender, 38
 hasChanged, 38
 initDataBase, 39
 initQuery, 39
 initTable, 39

initTags, 40
loadUserControls, 40
mline, 45
mreadStatus, 46
nextLine, 41
previousVals, 46
ptr1, 46
ptr2, 46
queryData, 41
readAny, 41
readData, 41, 42
readError, 42
readVal, 43, 44
tempVals, 46
tmpLine, 46
updateDb, 44
updateElements, 46
updateVals, 46
vdriftLimit, 46
ftpcGGridSender.cc, 94
ftpcGGridSender.cc
 __CLASS__, 94
 __METHOD__, 94
ftpcGGridSender.hh, 95
ftpcGGridSender.hh
 NUM_DB_ROWS, 95
ftpcGGridSender_i.cc, 96
ftpcGGridSender_i.cc
 __METHOD__, 96
ftpcHDLCTempsDaemon.cc, 97
ftpcHDLCTempsDaemon.cc
 runSender, 97
ftpcHDLCTempsSender, 47
 ftpcHDLCTempsSender, 48
ftpcHDLCTempsSender
 ~ftpcHDLCTempsSender, 48
 elementList, 55
 ftpcHDLCTempsSender, 48
 hasChanged, 48
 initDataBase, 49
 initQuery, 49
 initTable, 49
 initTags, 50
 loadUserControls, 50
 mline, 55
 mreadStatus, 56
 nextLine, 51
 pdriftLimit, 56
 previousVals, 56
 ptr1, 56
 ptr2, 56
 queryData, 51
 readAny, 51
 readData, 52
 readError, 53
 readVal, 53, 54
 tdriftLimit, 56
 tempVals, 56
 tmpLine, 56
 updateDb, 54
 updateElements, 56
 updateVals, 56
ftpcHDLCTempsSender.cc, 98
ftpcHDLCTempsSender.cc
 __CLASS__, 98
 __METHOD__, 98
ftpcHDLCTempsSender.hh, 99
ftpcHDLCTempsSender.hh
 NUM_DB_ROWS, 99
ftpcHDLCTempsSender_i.cc, 100
ftpcHDLCTempsSender_i.cc
 __METHOD__, 100
ftpcHDLCVoltagesDaemon.cc, 101
ftpcHDLCVoltagesDaemon.cc
 runSender, 101
ftpcHDLCVoltagesSender, 58
 ftpcHDLCVoltagesSender, 59
ftpcHDLCVoltagesSender
 ~ftpcHDLCVoltagesSender, 59
 elementList, 67
 ftpcHDLCVoltagesSender, 59
 hasChanged, 59
 initDataBase, 60
 initQuery, 60
 initTable, 61
 initTags, 61
 loadUserControls, 61
 mline, 67
 mreadStatus, 67
 nextLine, 62
 previousVals, 67
 ptr1, 67
 ptr2, 67
 queryData, 62
 readAny, 62
 readData, 63
 readError, 64
 readVal, 64, 65
 tempVals, 67
 tmpLine, 67
 updateDb, 66
 updateElements, 67
 updateVals, 67
 vdriftLimit, 68
ftpcHDLCVoltagesSender.cc, 102
ftpcHDLCVoltagesSender.cc
 __CLASS__, 102
 __METHOD__, 102

ftpcHDLCVoltagesSender.hh, 103
 ftpcHDLCVoltagesSender.hh
 NUM_DB_ROWS, 103
 ftpcHDLCVoltagesSender_i.cc, 104
 ftpcHDLCVoltagesSender_i.cc
 __METHOD__, 104
 ftpcTempsDaemon.cc, 105
 ftpcTempsDaemon.cc
 runSender, 105
 ftpcTempsSender, 69
 ftpcTempsSender, 70
 ftpcTempsSender
 ~ftpcTempsSender, 70
 elementList, 78
 extdriftLimit, 78
 ftpcTempsSender, 70
 hasChanged, 70
 initDataBase, 70
 initQuery, 71
 initTable, 71
 initTags, 72
 loadUserControls, 72
 mline, 78
 mreadStatus, 78
 nextLine, 72
 previousVals, 78
 ptr1, 78
 ptr2, 78
 queryData, 73
 readAny, 73
 readData, 73, 74
 readError, 74
 readVal, 75, 76
 tempVals, 78
 tmpline, 78
 updateDb, 76
 updateElements, 78
 updateVals, 78
 ftpcTempsSender.cc, 106
 ftpcTempsSender.cc
 __CLASS__, 106
 __METHOD__, 106
 ftpcTempsSender.hh, 107
 ftpcTempsSender.hh
 NUM_DB_ROWS, 107
 ftpcTempsSender_i.cc, 108
 ftpcTempsSender_i.cc
 __METHOD__, 108

 h2odriftLimit
 ftpcGasSystemSender, 35
 hasChanged
 ftpcAnodesSender, 6
 ftpcCathodeSender, 17

 ftpcGasSystemSender, 27
 ftpcGGridSender, 38
 ftpcHDLCTempsSender, 48
 ftpcHDLCVoltagesSender, 59
 ftpcTempsSender, 70

 initDataBase
 ftpcAnodesSender, 7
 ftpcCathodeSender, 17
 ftpcGasSystemSender, 27
 ftpcGGridSender, 39
 ftpcHDLCTempsSender, 49
 ftpcHDLCVoltagesSender, 60
 ftpcTempsSender, 70

 initQuery
 ftpcAnodesSender, 7
 ftpcCathodeSender, 18
 ftpcGasSystemSender, 28
 ftpcGGridSender, 39
 ftpcHDLCTempsSender, 49
 ftpcHDLCVoltagesSender, 60
 ftpcTempsSender, 71

 initTable
 ftpcAnodesSender, 8
 ftpcCathodeSender, 18
 ftpcGasSystemSender, 28
 ftpcGGridSender, 39
 ftpcHDLCTempsSender, 49
 ftpcHDLCVoltagesSender, 61
 ftpcTempsSender, 71

 initTags
 ftpcAnodesSender, 8
 ftpcCathodeSender, 19
 ftpcGasSystemSender, 29
 ftpcGGridSender, 40
 ftpcHDLCTempsSender, 50
 ftpcHDLCVoltagesSender, 61
 ftpcTempsSender, 72

 loadUserControls
 ftpcAnodesSender, 8
 ftpcCathodeSender, 19
 ftpcGasSystemSender, 29
 ftpcGGridSender, 40
 ftpcHDLCTempsSender, 50
 ftpcHDLCVoltagesSender, 61
 ftpcTempsSender, 72

 mline
 ftpcAnodesSender, 14
 ftpcCathodeSender, 24
 ftpcGasSystemSender, 35
 ftpcGGridSender, 45
 ftpcHDLCTempsSender, 55

ftpcHDLCVoltagesSender, 67
ftpcTempsSender, 78
mreadStatus
 ftpcAnodesSender, 14
 ftpcCathodeSender, 24
 ftpcGasSystemSender, 35
 ftpcGGridSender, 46
 ftpcHDLCTempsSender, 56
 ftpcHDLCVoltagesSender, 67
 ftpcTempsSender, 78

nextLine
 ftpcAnodesSender, 9
 ftpcCathodeSender, 19
 ftpcGasSystemSender, 30
 ftpcGGridSender, 41
 ftpcHDLCTempsSender, 51
 ftpcHDLCVoltagesSender, 62
 ftpcTempsSender, 72

NUM_DB_ROWS
 ftpcAnodesSender.hh, 83
 ftpcCathodeSender.hh, 87
 ftpcGasSystemSender.hh, 91
 ftpcGGridSender.hh, 95
 ftpcHDLCTempsSender.hh, 99
 ftpcHDLCVoltagesSender.hh, 103
 ftpcTempsSender.hh, 107

o2driftLimit
 ftpcGasSystemSender, 35

pdriftLimit
 ftpcHDLCTempsSender, 56

previousVals
 ftpcAnodesSender, 14
 ftpcCathodeSender, 24
 ftpcGasSystemSender, 35
 ftpcGGridSender, 46
 ftpcHDLCTempsSender, 56
 ftpcHDLCVoltagesSender, 67
 ftpcTempsSender, 78

ptr1
 ftpcAnodesSender, 14
 ftpcCathodeSender, 24
 ftpcGasSystemSender, 35
 ftpcGGridSender, 46
 ftpcHDLCTempsSender, 56
 ftpcHDLCVoltagesSender, 67
 ftpcTempsSender, 78

ptr2
 ftpcAnodesSender, 14
 ftpcCathodeSender, 24
 ftpcGasSystemSender, 35
 ftpcGGridSender, 46

ftpcHDLCTempsSender, 56
ftpcHDLCVoltagesSender, 67
ftpcTempsSender, 78

queryData
 ftpcAnodesSender, 9
 ftpcCathodeSender, 20
 ftpcGasSystemSender, 30
 ftpcGGridSender, 41
 ftpcHDLCTempsSender, 51
 ftpcHDLCVoltagesSender, 62
 ftpcTempsSender, 73

readAny
 ftpcAnodesSender, 10
 ftpcCathodeSender, 20
 ftpcGasSystemSender, 30
 ftpcGGridSender, 41
 ftpcHDLCTempsSender, 51
 ftpcHDLCVoltagesSender, 62
 ftpcTempsSender, 73

readData
 ftpcAnodesSender, 10
 ftpcCathodeSender, 20, 21
 ftpcGasSystemSender, 30, 31
 ftpcGGridSender, 41, 42
 ftpcHDLCTempsSender, 52
 ftpcHDLCVoltagesSender, 63
 ftpcTempsSender, 73, 74

readError
 ftpcAnodesSender, 11
 ftpcCathodeSender, 21
 ftpcGasSystemSender, 31
 ftpcGGridSender, 42
 ftpcHDLCTempsSender, 53
 ftpcHDLCVoltagesSender, 64
 ftpcTempsSender, 74

readVal
 ftpcAnodesSender, 11, 12
 ftpcCathodeSender, 21–23
 ftpcGasSystemSender, 32, 33
 ftpcGGridSender, 43, 44
 ftpcHDLCTempsSender, 53, 54
 ftpcHDLCVoltagesSender, 64, 65
 ftpcTempsSender, 75, 76

runSender
 ftpcAnodesDaemon.cc, 81
 ftpcCathodeDaemon.cc, 85
 ftpcGasSystemDaemon.cc, 89
 ftpcGGridDaemon.cc, 93
 ftpcHDLCTempsDaemon.cc, 97
 ftpcHDLCVoltagesDaemon.cc, 101
 ftpcTempsDaemon.cc, 105

tdriftLimit
 ftpcHDLCTempsSender, 56

tempVals
 ftpcAnodesSender, 14
 ftpcCathodeSender, 24
 ftpcGasSystemSender, 35
 ftpcGGridSender, 46
 ftpcHDLCTempsSender, 56
 ftpcHDLCVoltagesSender, 67
 ftpcTempsSender, 78

tmpline
 ftpcAnodesSender, 14
 ftpcCathodeSender, 25
 ftpcGasSystemSender, 35
 ftpcGGridSender, 46
 ftpcHDLCTempsSender, 56
 ftpcHDLCVoltagesSender, 67
 ftpcTempsSender, 78

updateDb
 ftpcAnodesSender, 13
 ftpcCathodeSender, 23
 ftpcGasSystemSender, 33
 ftpcGGridSender, 44
 ftpcHDLCTempsSender, 54
 ftpcHDLCVoltagesSender, 66
 ftpcTempsSender, 76

updateElements
 ftpcAnodesSender, 14
 ftpcCathodeSender, 25
 ftpcGasSystemSender, 35
 ftpcGGridSender, 46
 ftpcHDLCTempsSender, 56
 ftpcHDLCVoltagesSender, 67
 ftpcTempsSender, 78

updateVals
 ftpcAnodesSender, 14
 ftpcCathodeSender, 25
 ftpcGasSystemSender, 35
 ftpcGGridSender, 46
 ftpcHDLCTempsSender, 56
 ftpcHDLCVoltagesSender, 67
 ftpcTempsSender, 78

vdriftLimit
 ftpcAnodesSender, 15
 ftpcCathodeSender, 25
 ftpcGGridSender, 46
 ftpcHDLCVoltagesSender, 68